

NIKE3D

A Nonlinear, Implicit, Three-Dimensional Finite Element Code For Solid And Structural Mechanics User's Manual

**Bradley N. Maker
Methods Development Group
Mechanical Engineering Department**

Robert M. Ferencz, consultant

John O. Hallquist, originator

April 14, 1995

NIKE3D

A Nonlinear, Implicit, Three-Dimensional Finite Element Code For Solid And Structural Mechanics User's Manual

**Bradley N. Maker
Methods Development Group
Mechanical Engineering Department**

**Robert M. Ferencz
consultant**

**John O. Hallquist
originator**

April 14, 1995

NIKE3D:
A Nonlinear, Implicit, Three-Dimensional Finite Element Code
For Solid And Structural Mechanics - User's Manual

ABSTRACT

This report provides a user's manual for NIKE3D, a fully implicit three-dimensional finite element code for analyzing the finite strain static and dynamic response of inelastic solids, shells, and beams. Spatial discretization is achieved by the use of 8-node solid elements, 2-node truss and beam elements, and 4-node membrane and shell elements. Over twenty constitutive models are available for representing a wide range of elastic, plastic, viscous, and thermally dependent material behavior. Contact-impact algorithms permit gaps, frictional sliding, and mesh discontinuities along material interfaces. Several nonlinear solution strategies are available, including Full-, Modified-, and Quasi-Newton methods. The resulting system of simultaneous linear equations is either solved iteratively by an element-by-element method, or directly by a factorization method, for which case bandwidth minimization is optional. Data may be stored either in or out of core memory to allow for large analyses.

PREFACE

NIKE3D has been used at the Lawrence Livermore National Laboratory over the past fourteen years to study the static, quasi-static, and dynamic response of structures undergoing finite deformations. Available to the public, NIKE3D's use outside the Laboratory has grown rapidly as well. In 1989 the originator of NIKE3D, Dr. John O. Hallquist, left the Laboratory to pursue a career with his private company. The author wishes to emphasize that Dr. Hallquist was responsible for developing the kernel of code from which the current version is evolving. His genius at the Laboratory will be sorely missed.

Since its introduction in 1980 numerous improvements have been made to NIKE3D (Hallquist [1984]). Perhaps the most significant development has been the addition of structural elements including trusses, beams, membranes, and shells to augment the 8-node solid element. The first structural element was added by Slater [1982], who implemented a shell element that was a product of his Ph.D. research. At this time NIKE3D was extended and reorganized to include rotational degrees of freedom and "ports" for new element classes. Although Slater's shell was used successfully in many calculations, it has been dropped from NIKE3D in favor of a shell element based on the work of Hughes and coworkers [1981a-d] which, unlike the previous implementation, is valid for finite strains. In a straightforward exercise this shell has been degenerated into a rectangular beam element. Truss and membrane elements are obtained from the beam and shell elements, respectively, by one point integration in the thickness direction while ignoring transverse shear.

Many organizational changes have been made to reduce the number of I/O operations that often dominate the solution cost. The element overlays have been recoded for vectorization. The Green-Naghdi stress rate (Johnson and Bammann [1984]) has been adopted in favor of the Jaumann rate for the solid elements to obtain improved behavior for kinematic hardening in the plasticity model. However, the Jaumann rate has been retained for the structural elements. The implementation of radial return plasticity (Krieg and Key [1976]) has eliminated the very costly and less accurate subincrementation algorithm. Plane stress structural elements also use radial return with a simple additional iteration to uniquely solve for the normal strain. A stiffness matrix formulation of Hughes [1980] has been adopted that is consistent with the constant pressure assumption (Nagtegaal, Parks, and Rice [1974]) in the solid elements. The sliding interface algorithm has also been extended to shell elements and modified to increase its reliability and accuracy.

TABLE OF CONTENTS

ABSTRACT	I
PREFACE.....	II
1. INTRODUCTION	
1.1 RECENT DEVELOPMENTS.....	1-2
2. OVERVIEW OF NIKE3D	
2.1 SOLUTION PROCEDURE.....	2-2
2.2 ELEMENT LIBRARY.....	2-2
2.3 MATERIAL MODELS.....	2-4
2.4 NONLINEAR SOLUTION ALGORITHMS	2-5
2.5 LINEAR SOLVERS.....	2-5
2.6 SLIDING INTERFACES	2-6
2.7 NATURAL FREQUENCY AND MODE SHAPE ANALYSIS.....	2-8
2.8 CAPACITY.....	2-9
3. EXECUTION	
3.1 COMMAND LINE OPTIONS.....	3-1
3.2 INPUT FORMAT CONVERSION.....	3-2
3.3 SENSE SWITCH CONTROLS	3-3
3.4 RUNTIME ERRORS	3-4
4. INPUT FORMAT	
4.1 CONTROL DECK.....	4-1
CONTROL DECK NOTES	4-11
4.2 MATERIAL DECK.....	4-17
TYPE 1 (ELASTIC).....	4-19
TYPE 2 (ORTHOTROPIC ELASTIC).....	4-20
TYPE 3 (ELASTIC-PLASTIC).....	4-23
TYPE 4 (THERMO-ELASTIC-PLASTIC).....	4-25
TYPE 5 (SOIL AND CRUSHABLE FOAM).....	4-27
TYPE 6 (VISCOELASTIC)	4-29
TYPE 7 (THERMO-ORTHOTROPIC ELASTIC).....	4-30
TYPE 8 (THERMO-ELASTIC-CREEP)	4-31
TYPE 9 (POWER LAW PLASTICITY).....	4-33
TYPE 10 (POWER LAW THERMO-ELASTIC PLASTIC).....	4-34
TYPE 11 (TRANSIENT THERMAL CREEP).....	4-36

TYPE 12 (RAMBERG-OSGOOD ELASTIC-PLASTIC).....	4-38
TYPE 13 (GENERAL ANISOTROPIC THERMO-ELASTIC)	4-40
TYPE 14 (ORIENTED BRITTLE DAMAGE).....	4-41
TYPE 15 (MOONEY-RIVLIN RUBBER).....	4-43
TYPE 16 (THERMO-PLASTIC MELT).....	4-44
TYPE 17 (FOUNDATION BOUNDARY SPRING).....	4-48
TYPE 18 (TRANSVERSELY ISOTROPIC HYPERELASTICITY)	4-49
TYPE 19 (STRAIN RATE SENSITIVE POWER LAW PLASTICITY)....	4-51
TYPE 20 (RIGID BODY)	4-52
TYPE 23 (THERMO-ORTHOTROPIC ELASTIC LAMINATE).....	4-54
CROSS SECTION DEFINITIONS - BEAM AND TRUSS	4-56
CROSS SECTION DEFINITIONS - MEMBRANE AND SHELL	4-58
4.3 USER-DEFINED INTEGRATION RULES FOR BEAMS.....	4-60
4.4 USER-DEFINED INTEGRATION RULES FOR SHELLS.....	4-61
4.5 NODE POINT DECK.....	4-62
4.6 HEXAHEDRON ELEMENT DECK.....	4-63
4.7 BEAM ELEMENT DECK.....	4-64
4.8 SHELL ELEMENT DECK.....	4-66
4.9 RIGID NODE AND FACET DECK	4-67
4.10 DISCRETE ELEMENT DECK.....	4-68
DISCRETE ELEMENT CONTROL CARD.....	4-68
DISCRETE ELEMENT MATERIAL DECK.....	4-68
TYPE 1 (LINEAR ELASTIC)	4-69
TYPE 2 (LINEAR VISCOUS).....	4-69
TYPE 3 (ISOTROPIC ELASTOPLASTIC).....	4-69
TYPE 4 (NONLINEAR ELASTIC).....	4-69
TYPE 5 (NONLINEAR VISCOUS).....	4-70
TYPE 6 (GENERAL NONLINEAR).....	4-70
DISCRETE SPRING AND DAMPER DEFINITIONS	4-71
DISCRETE MASS DEFINITIONS.....	4-71
4.11 CONSTRAINED NODE PAIRS DECK.....	4-72
4.12 1D SLIDELINE DECK.....	4-73
4.13 SLIDING SURFACE DECK.....	4-76
4.14 NODAL REORDER DATA.....	4-81
4.15 STONEWALL AND SYMMETRY PLANE DECK.....	4-82
4.16 NODE PRINT BLOCKS.....	4-86

4.17 SOLID ELEMENT PRINT BLOCKS.....	4-86
4.18 BEAM ELEMENT PRINT BLOCKS.....	4-87
4.19 SHELL ELEMENT PRINT BLOCKS.....	4-87
4.20 LOAD CURVE DECK	4-88
4.21 CONCENTRATED NODAL LOADS DECK	4-89
4.22 TEMPERATURE PROFILE DECK.....	4-90
4.23 PRESSURE BOUNDARY CONDITION DECK.....	4-91
4.24 DISPLACEMENT BOUNDARY CONDITION DECK	4-93
4.25 BASE ACCELERATION BODY FORCE LOADS DECK.....	4-94
4.26 ANGULAR VELOCITY BODY FORCE LOADS DECK.....	4-95
4.27 INITIAL VELOCITY DECK.....	4-96
4.28 AERODYNAMIC DRAG LOAD DECK.....	4-97
4.29 FOUNDATION NODE BOUNDARY CONDITION	4-99
5. RESTART INPUT FORMAT	
5.1 RESTART CONTROL DECK.....	5-2
5.2 ANALYSIS TYPE CHANGE DECK.....	5-4
5.3 LOAD CURVE REDEFINITION DECK.....	5-4
5.4 SLIDING INTERFACE REDEFINITION DECK.....	5-5
6. EXAMPLES	
6.1 BAR IMPACTING RIGID WALL.....	6-2
6.2 PIPE WHIP	6-5
6.3 DYNAMIC RESPONSE OF OVERPASS BRIDGE (D.B. McCallen).....	6-8
6.4 FOOTED COLUMN	6-11
6.5 ELASTIC-PLASTIC TUBE (S.A. Perfect)	6-14
6.6 VOID GROWTH IN TENSILE SPECIMEN (P.J. Raboin)	6-15
6.7 HUMAN KNEE JOINT.....	6-18
7. ACKNOWLEDGMENTS	
8. REFERENCES	

1. INTRODUCTION

NIKE3D is a general purpose, three dimensional, nonlinear finite element code for solving problems in solid and structural mechanics. It utilizes implicit time integration, making it most efficient for static or low rate dynamic problems. For high rate dynamic problems such as high speed crash and impact, Lawrence Livermore's explicitly integrated DYNA3D code (Whirley and Hallquist, [1991]) is more appropriate.

NIKE3D utilizes a relatively small set of elements. All elements use low order interpolation, requiring no midside node definitions. This approach chooses highly efficient elements over more costly higher order elements (which have been shown to be more accurate in *linear* analyses.) Using many efficient elements, rather than few higher order elements, the analyst may more accurately describe the complex geometries typical of large finite element models.

Nonlinearities may be introduced to an analysis in several ways. Geometric nonlinearities involve large deformations and rotations, where the "small angle" and "constant area" assumptions of linear theories are invalid. Contact between two bodies introduces a nonlinear process when separation is allowed and contact areas change. More obviously, material behaviors such as plasticity, creep, and thermal distortion introduce nonlinearities. NIKE3D allows the analyst to consider these and other nonlinear phenomenon.

Analyses with NIKE3D typically concern loadings applied in several increments, or steps, to accurately resolve geometric or material nonlinearities. Within each step, a nonlinear solver drives the iterative process used to satisfy equilibrium. Within each equilibrium iteration, a system of linear equations must be solved. Several nonlinear and linear solvers are available.

For dynamic problems, load steps represent increments in time. In *quasi-static* analysis, these steps are increments in *quasi-time*, a monotonically increasing time-like parameter which characterizes the evolution of the loading. The unconditional stability of the implicit time integrator allows the user to specify the time step size, its choice being governed by convergence and accuracy considerations. Proper choice of time step size, where convergence is maintained, accuracy is maximized, and computational cost is minimized, represents one of the most difficult challenges to the analyst using implicit finite element methods. A variety of diagnostic and automatic time step control tools are available to simplify this task.

1.1 Recent Developments

Since the 1991 manual was released, development of NIKE3D has proceeded steadily. Major direction has come from problems in civil engineering, such as analysis of seismically driven reinforced concrete structures, and from metal forming applications. The emergence of high powered desktop computing, especially advances in memory availability, has driven development of the workstation versions of NIKE3D. These include Sun, SGI, IBM, and Hewlett-Packard workstations, but can easily be extended to most UNIX-based machines. The code still operates in single precision, creating numerical precision difficulty for some simulations on 32-bit machines, but reducing memory requirements by nearly a factor of two. A Cray UNICOS version provides 64-bit precision for verification of workstation results, and large scale production analyses.

Important enhancements have been made to NIKE3D's contact capability. The search strategy for general sliding interfaces has been enhanced to improve accuracy for problems with large relative motion in a time step. A death/burial option allows interfaces to be removed from a calculation, such as for removal of tooling in springback calculations. A new contact interface, the 1D slideline, has been added to treat the interaction of beam elements with solid elements, such as between rebar and concrete. The method of augmented Lagrangians has been added as an alternative to the penalty formulation for many of NIKE3D's contact interfaces. This iterative method allows the user to specify an accuracy requirement which contact constraints must satisfy, which is independent of the penalty parameter.

NIKE3D's material model suite has increased by over 50%. Geologic and concrete materials may be treated using the 3D Ramberg-Osgood model, or the Oriented Brittle Damage model, which includes a failure mechanism that dissipates energy in a mesh-independent manner. Thermal analysis capability has been extended by introducing a material reference temperature, a significant bug fix in the thermo-elastic-plastic model, thermo-elastic beams, and the new thermo-plastic melt model. Metal forming analyses may utilize the new strain rate sensitive plasticity model for shell elements, and the rigid material for tooling and obstacle representation. The new hyperelastic and transversely isotropic hyperelastic models are useful for rubber and biological soft tissue applications. All of the materials (except rigid) allow optional inclusion of Rayleigh damping based on their elastic properties.

The output data NIKE3D produces has been enhanced to include reaction forces at nodes and rigid bodies whose displacements are prescribed. In addition, total reaction forces across each contact

interface are now computed and included in the printed output and plot database. Printing of node and element data can be suppressed without defining print blocks. And data displayed to the terminal screen has been reformatted and enhanced to provide more useful information as NIKE3D executes.

To aid in developing and debugging models, many new diagnostic tests and messages have been included in the initialization phase. Two new interactive sense switches have been added to allow intermediate, "unconverged" solutions to be viewed using the post processor. This allows visual inspection of a deformed mesh for model defects, and helps to identify sources of convergence difficulty. The rigid material can also be used to selectively "rigidize" portions of the model, speeding computation during the debugging process.

The greatest challenge in analysis with NIKE3D is still selection of the time step size. To aid in monitoring the progress of the nonlinear solver, the screen output has been enhanced. Convergence may be controlled using a new residual norm, and/or by a new displacement tolerance whose value does not change as the total deformation in the problem increases. Most importantly, automatic time step control has matured, trapping and recovering from errors that previously generated floating point exceptions.

Finally, many small bug fixes and improvements have been made to enhance NIKE3D's overall coherence, simplicity, and utility. Much of this work was motivated or developed by members of the Methods Development Group's collaborators program. We continue to hope that improvements made by others will be made available to the author for possible incorporation into the public domain version.

2. OVERVIEW OF NIKE3D

The following is a general discussion of the capabilities of NIKE3D. Formal documentation of the theoretical aspects of NIKE3D is not yet available in a single text, however journal citations are included which provide detailed information on much of the formulation. Other references include the theory section of the NIKE2D manual (Engelmann and Hallquist, [1990]), the Theoretical Manual for DYNA3D (Hallquist [1983]) and in the DYNA3D Course Notes (Hallquist [1987]).

Capabilities of the 1994 version of NIKE3D include:

- large deformation, large strain analyses
 - eigenvalue extraction with frequency shift
 - thermal stresses, time varying thermal profiles
 - eight node continuum elements
 - four node shell and two node beam/truss structural elements
 - two node discrete (1D) elements
 - a wide variety of elastic and inelastic material models
 - full-, modified-, and quasi-Newton nonlinear solvers
 - arc length nonlinear solver
 - bandwidth minimization
 - automatic time step control
 - in-core vectorized direct linear equation solver
 - out-of-core profile direct linear equation solver
 - Element-By-Element vectorized iterative linear equation solver
 - user-defined cross section integration rules for structural elements
 - constrained node pairs
 - 1D slidelines
 - tied and sliding interfaces
 - concentrated nodal loads
 - displacement, pressure boundary conditions
 - body force loads due to base acceleration, angular velocity
 - initial velocities
 - aerodynamic drag boundary conditions
 - general restart capability
-

- selective node and element output printing
- binary database generation for TAURUS post processor
- stress initialization for DYNA3D explicit dynamic analysis
- large capacity (limited only by available memory)

2.1 Solution Procedure

NIKE3D is based on an updated Lagrangian formulation. During each load step, NIKE3D computes the nodal displacement increments which produce a geometry that satisfies equilibrium at the end of the step. This involves the solution of a set of (potentially) nonlinear equations. Several nonlinear solvers based on the Newton method are available. The nonlinear solution process in each step is iterative. Equilibrium is obtained when one or more user defined convergence tolerances are met.

During each equilibrium iteration of the nonlinear solver, NIKE3D recomputes internal and external forces and the global stiffness matrix (if necessary) using the current estimate of geometry. These forces are combined to form the right hand side or residual vector. The system of linear equations $[\mathbf{K}]\{\mathbf{u}\} = \{\mathbf{R}\}$ formed by the stiffness matrix $[\mathbf{K}]$ and the right hand side vector $\{\mathbf{R}\}$ is then solved to produce a new displacement increment $\{\mathbf{u}\}$. Solution of this linear system consumes a majority of the computational effort in a large NIKE3D analysis. Both direct and iterative linear solvers are available.

After obtaining updated displacement increments, the displacement, energy, and residual norms are computed, and equilibrium convergence is tested using user defined tolerances. These norms are optionally printed to the terminal so that the user can monitor the progress of the equilibrium iteration process. Once convergence is obtained, displacements and stresses are updated, output is optionally generated, and NIKE3D proceeds with the next load step. If convergence is not achieved within the user-specified iteration limits, the optional automatic time step controller will adjust the step size and try again.

2.2 Element Library

Spatial discretization is accomplished with solid, beam, and/or shell elements (Fig. 1). Eight node solid elements are integrated with a 2x2x2 point Gauss quadrature rule. The B-bar (Hughes, [1980]) and incompatible modes (Taylor, [1976]) formulations are used to avoid locking and

enhance bending performance. The implementation allows these to degenerate to six node solid elements. Experience has shown the degenerated elements to be inaccurate, and their use should be avoided whenever possible. As yet, tetrahedral elements are not available in NIKE3D due to their poor performance in large deformation, nonlinear applications.

Four node shell elements use 2x2 Gauss integration in the plane, and one of many available schemes for integration through the thickness. The Hughes-Liu shell (Hughes, [1981b]) is a degenerate formulation which provides finite strain and thinning behavior, while the co-rotational YASE shell (Engelmann, [1989]) utilizes enhanced strain fields. Two node beam elements use one integration point along the length, with many options for integration of the cross section. The structural elements also accept varying nodal thicknesses and user specified cross section integration rules. The latter capability is particularly useful with the laminated composite shell material model.

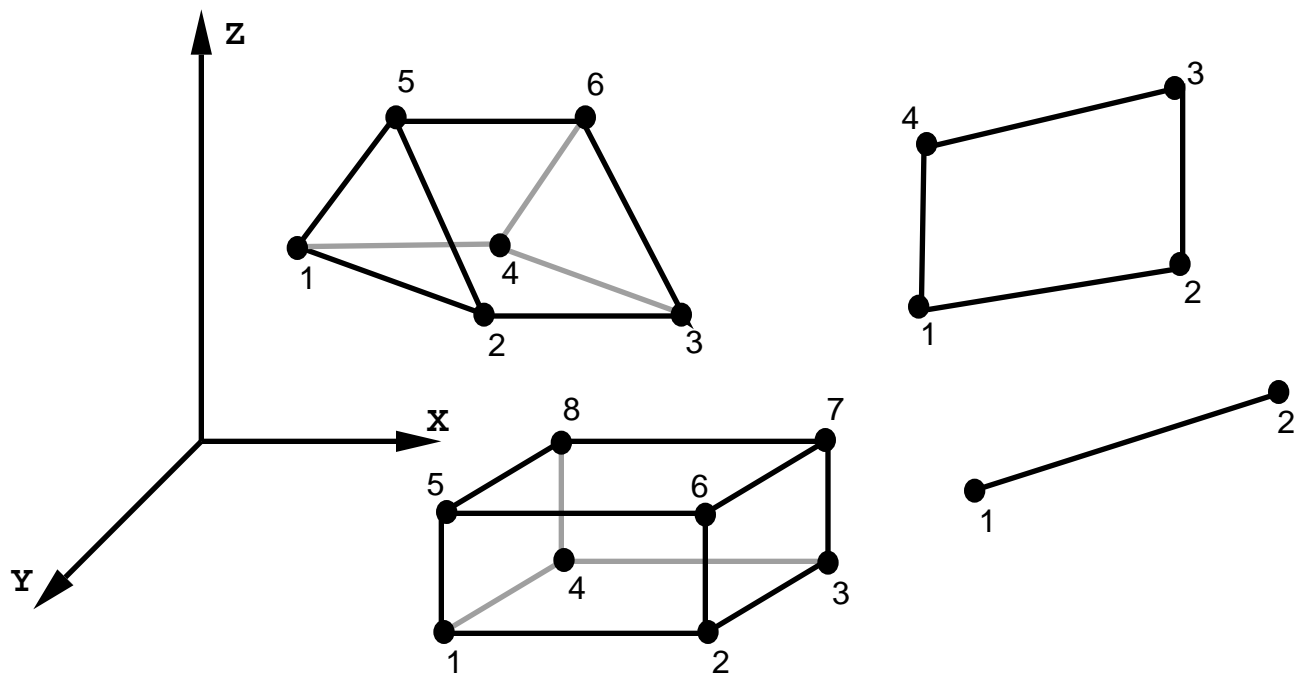


Figure 2-1. Elements in NIKE3D.

2.3 Material Models

A majority of the material models in NIKE3D are based on incremental formulations. In the solid elements, these materials use the Green-Naghdi stress rate to integrate stresses objectively (Johnson and Bammann [1984]). The Jaumann rate is used in the structural elements. The Mooney-Rivlin rubber model is based on a hyperelastic formulation, where the total stress is computed each step.

Material properties are defined by assigning a material identification number to each element. These numbers identify material datasets, which include the model type, coefficients, and cross section properties. In this manner, elements in the same physical part may be assigned unique material numbers (for display purposes), and material models may be used with more than one set of coefficients. Material models are available in NIKE3D to represent the following material behaviors:

- elastic
- orthotropic elastic
- elastic-plastic
- thermo-elastic-plastic
- soil and crushable foam
- linear viscoelastic
- thermo-orthotropic elastic
- thermo-elastic-creep
- power law elastic-plastic
- power law thermo-elastic-plastic
- transient thermal creep
- Ramberg-Osgood elastic-plastic
- general anisotropic thermo-elasticity
- oriented brittle damage
- Mooney-Rivlin rubber
- thermo-plastic melt
- transversely isotropic hyperelastic
- strain rate sensitive power law elastic-plastic
- rigid
- thermo-orthotropic elastic laminate

2.4 Nonlinear Solution Algorithms

The nonlinear solver is responsible for finding a set of nodal displacements which satisfy equilibrium at each load step. Several iterative nonlinear solvers are implemented in NIKE3D. The full Newton method recomputes and refactorizes the global stiffness matrix at each iteration. The modified Newton method reforms the global stiffness matrix after a user specified number of iterations. The popular quasi-Newton methods employ an inexpensive, approximate update to the factorized matrix each iteration. Several updates are available, including BFGS , Broyden, Davidon, and Davidon-Fletcher-Powell (Mathies and Strang [1979], Bathe and Cimento [1980], Walker [1979]). Generally, the quasi-Newton methods require fewer iterations than the modified Newton-Raphson method since unlike the latter method, they have superlinear local convergence. A line search is available with each of these schemes, along with an automated stiffness reformation scheme to avoid divergence.

By default, NIKE3D uses the BFGS method. An extensive set of diagnostic messages have been incorporated into the quasi-Newton solvers to allow their convergence progress to be monitored. By setting the *number of equilibrium iterations between stiffness reformations* to one, the full Newton with line search method is recovered using the BFGS option.

2.5 Linear Solvers

Within the iteration loop of the nonlinear solution algorithm, a linearized system of simultaneous equations of the form $[\mathbf{K}]\{\mathbf{u}\} = \{\mathbf{R}\}$ must be solved. Two linear solution schemes are now available, a direct method and an iterative method.

The direct linear solution method (Taylor, Wilson, and Sackett [1980], Storaasli, Nguyen, and Agarwal [1990]) requires the formation of a global stiffness matrix. The bandwidth minimizer (Sackett [1977], Gibbs, Poole, and Stockmeyer [1976]) optionally reorders equations to generate an optimum profile, reducing storage requirements in a user-transparent fashion. The stiffness matrix is then factorized and stored in profile form either in core memory, or on disk. Successive back-substitutions required in the quasi-Newton algorithms may then be performed inexpensively using this factorized matrix. However, for an $N \times N \times N$ element mesh, storage and I/O costs for this scheme increase proportional to N^5 , and rapidly dominate the cost of the analysis. These memory requirements limit the size of feasible models, especially on today's smaller workstations.

The development and implementation of the Element-By-Element (EBE) iterative solver by Ferencz [1989] offers an alternative to this dilemma. The preconditioned conjugate gradient method is used

to obtain an approximate solution to the linear system. The approximation is improved iteratively until a user specified tolerance is met. This method avoids the factorization of the global stiffness matrix at the expense of evaluating (over perhaps many iterations) matrix-vector products involving **K**. These matrix operations are performed at the element level, hence the global stiffness matrix need not be assembled. To speed convergence of the iterative procedure, several preconditioners are available, including diagonal, Crout, Choleski, and Gauss-Seidel. The auxiliary matrix-vector operations involved in preconditioning are also performed at the element level, rendering the entire process Element-By-Element.

2.6 Sliding Interfaces

NIKE3D's multi-body contact algorithms are based on a master - slave approach. Typically, one side of a potential interface is identified as the "master" side, the other side the "slave". Each side is identified in the input deck as an arbitrarily ordered list of facets. These facets are typically element faces. For solid element faces, the algorithm automatically computes the outward direction to the surface. For shell element faces, the outward direction is taken from the node numbering sequence of each facet. Three node triangular facets are allowed. Four node quadrilateral facets need not be planar.

Internal logic identifies a master facet for each slave node and a slave facet for each master node. This information is updated each time step as the slave and master nodes slide along their respective surfaces. In this manner, arbitrarily large relative motion between the surfaces is allowed.

NIKE3D computes a total reaction force across each interface. This data is included in the printed and plotted output. The forces are expressed as vector components in the global coordinate system. They may be viewed in phase two of TAURUS with the `stime` command, using component numbers generated by NIKE3D. These numbers uniquely identify global force components for each interface in the model, hence they vary from model to model. NIKE3D includes a tabulated list of component numbers in the screen output and in the printed output file.

Four types of interfaces can presently be defined including:

- tied.
- sliding only,
- sliding with gaps and friction
- single surface

The **tied** slide surface requires that both master and slave sides be initially in contact. The algorithm prevents all relative motion between the slave nodes and their respective master facets. This treatment is not symmetric with respect to slave and master designation. *The surface which is more coarsely zoned should be chosen as the master surface.* The tied slide surface is useful to enforce compatibility across mesh discontinuities, since element boundaries need not be aligned across the interface. Overlapping tied surfaces are treated by releasing slave nodes that extend beyond the edges of the master surface.

The **sliding only** slide surface also requires that both master and slave sides be initially in contact. This algorithm allows only tangential motion between the surfaces. All motion in the normal direction is suppressed, potentially generating both tensile and compressive normal stresses across the interface. This treatment is not symmetric with respect to slave and master designation. *The surface which is more coarsely zoned should be chosen as the master surface.* The sliding only slide surface is useful in modeling frictionless contact without separation.

The **sliding with gaps and friction** slide surface provides general multibody contact capability. The surfaces need not be initially in contact. Slide surface nodes that are defined to be initially penetrated are moved back onto the appropriate surface prior to execution. During execution, the surfaces may close, open, and separate, evolving the contact area as necessary to satisfy global equilibrium. Treatment is symmetric, hence the master - slave designation is arbitrary. Both static and kinetic friction coefficients may be specified, with an exponential decay capability for transition between the two regimes.

The **single surface** option allows parts to fold onto themselves. In this option, only the slave side of the interface is defined. This is often useful for buckling analysis, where potential contact areas are difficult to determine a priori. Due to the complex nature of this option, its use should be avoided in favor of the master-slave algorithm whenever possible. The single surface option is currently under development. The author encourages users to communicate their experiences.

During initialization, NIKE3D checks for interpenetration between master and slave surfaces in the initial geometry, and adjusts nodal coordinates if necessary to return nodes to their opposing surfaces. Similarly, gaps are eliminated between tied and sliding only type interfaces. Node relocation information is included in the printed output file.

NIKE3D's contact algorithms are based on the penalty method, where penalty "springs" are automatically generated between contact surfaces when interpenetration is detected. These springs produce contact forces that are proportional to interpenetration depth. The drawback to this method is that some interpenetration of opposing surfaces must occur to develop interface forces. Ideally, by choosing relatively stiff penalty springs, the interpenetration can be reduced to insignificant values. To achieve this, NIKE3D employs a complex algorithm to compute the default penalty stiffness which depends on material and mesh properties.

A penalty stiffness scale factor is available to adjust the default stiffness of the contact interface. A scale factor of less than one reduces penalty stiffness, allowing more interpenetration but (typically) leading to faster equilibrium convergence. A scale factor greater than one reduces interpenetration, but can lead to poorly conditioned numerical systems and convergence failure. As a general guideline, a scale factor of 0.1 to 0.01 may be used if equilibrium convergence is slow in a contact dominated problem. The increased interpenetration which results should be observed using TAURUS to determine its significance in a particular model.

The augmented Lagrangian contact formulation has proven to be very effective for many contact dominated problems. The method is iterative, and is applied after convergence is obtained with the standard penalty method. A fixed number of augmentations may be requested, or the force or gap norms may be used to control convergence of the time step based on accuracy of the contact constraint. After each augmentation, the time step is repeated and a new equilibrium configuration is determined. Convergence of the augmentation procedure assures the accuracy of the contact constraint to within the user specified tolerance, independent of penalty stiffness value. Progress of the augmentation loops may be monitored by activating sense switch SW3.

Since the augmented Lagrangian method iterates to enforce contact, a large penalty stiffness is not required to eliminate interpenetration normally associated with the penalty method. Using a relatively small penalty stiffness often allows equilibrium convergence to be obtained more rapidly between augmentations, offsetting the cost of the additional augmentation loop. For details of the NIKE3D implementation, see Laursen and Maker [1995].

2.7 Natural Frequency and Mode Shape Analysis

The eigenvalue capability in NIKE3D is based on an incore eigenvalue program developed by Taylor [1979]. Eigenvalues are computed (with the stiffness matrix and eigenvectors stored out-

of-core) by the subspace iteration method as described by Bathe and Wilson [1976]. A shift option has been added to analyze higher modes more inexpensively. The computed eigenvectors are normalized and written into the TAURUS binary database as plot states. TAURUS may be used to view the mode shapes, with frequency displayed as the "time" value of each state. By sequentially increasing and decreasing the displacement scale factor, TAURUS can display "vibration" of the model in each mode.

2.8 Capacity

NIKE3D's model size limit is machine dependent. On Cray machines, memory allocation is dynamic, and the stiffness, element constitutive data, and BFGS update vectors may be optionally stored on disk. This architecture yields the flexibility to consume as much of the computer's resources as the user wishes. At LLNL, "large" analyses contain on the order of 10,000 elements. Much larger analyses have been conducted.

On VAX and workstation machines, memory allocation is currently static, hence model size is bounded. On these machines, it is especially important to keep the entire model in core memory. Usage of disk files for scratch storage of element and stiffness matrix data is optional, but discouraged due to severe performance penalty. The default size requires roughly 20 Mbytes of memory, and accommodates models of the order of 1,000 elements. This limit can be easily modified by recompiling two subroutines and rebuilding the executable code.

3. EXECUTION

NIKE3D contains virtually no mesh generation capability. At LLNL, the three-dimensional mesh generators INGRID (Stillman and Hallquist [1985]) and SLIC (Gerhard [1979]) provide preprocessing capability, and support most of NIKE3D's options. Occasionally, users must edit the preprocessor-generated input deck directly, especially to invoke new features.

The instructions below detail execution procedures and run-time interrogation of NIKE3D. Once the analysis has completed, results are available in printed form as well as in a plotfile database. NIKE3D generates binary plot and restart databases in *familied* file sets. These files have a common base name of up to six characters, followed by a two digit family number. The user may select the base names, or use defaults provided. The interactive three dimensional color graphics postprocessors TAURUS (Spelce, Hallquist, and Brown [1990]) and GRIZ (Dovey and Spelce [1993]) may be used to display results from the plot database. Graphical postprocessing is the principal means of examining the results of a NIKE3D analysis.

In this manual the following typeface conventions are adopted: **Bold** type indicates commands entered by the user; *italic* type indicates arguments for which the user substitutes values; and typewriter indicates verbatim names, values, or responses.

3.1 Command Line Options

The execution line for NIKE3D is:

NIKE3D **I**=*inf* **O**=*otf* **G**=*ptf* **D**=*dpf* **T**=*tpf* **F**=*rdf* **S**=*stf* **C**=*cnf*

where

inf = input file,

otf = high speed printer file (default basename = n3hsp)

ptf = binary plot file for graphics (default basename = n3plot)

dpf = dump file for restarting analysis (default basename = n3dump)

tpf = TOPAZ3D (Shapiro [1985]) temperature file

rdf = running restart file (default basename = n3runr)

stf = stress initialization file for DYNA3D (written after completion of final step)

cnf = convergence information from EBE iterative solver (default name = itrcont)

File names must be unique and can have up to six characters. Dump files accumulate as an analysis proceeds, whereas running restart files are overwritten after each time step completes. A new family member in the plot file database is created whenever a dump file is generated. This partition in the database allows portions to be selectively copied and archived during parameter studies. If a stress initialization filename is not specified on the command line, no file is generated.

When restarting an analysis from a dump file or running restart file, the execution line becomes

NIKE3D I=inf O=otf G=ptf D=dpf R=rtf T=tpf F=rdf

where

inf = restart input file described at end of this manual (optional).

rtf = eight character dump file or running restart file name.(ie. n3dump01)

File name dropouts are permitted, for example

NIKE3D I=inf

NIKE3D R=rtf

are valid execution lines. Omitted filenames will default to those used to begin the original analysis. When NIKE3D terminates, all scratch files (diskx, etc.) are destroyed. The restart file, plot files, and high-speed printer files remain on disk. Of these, only the restart file is needed to continue the interrupted analysis.

3.2 Input Format Conversion

NIKE3D has an input format conversion option for use converting old (nn, nr, en) format input files to the new (f1) format. To convert an old file, first change its name to "convert". Then execute NIKE3D using the command line:

NIKE3D I=convert, C=nwf

where

nwf = new input file in f1 format (generated by NIKE3D)

The contents of *convert* will be echoed into the file *nwf* using the f1 format, and NIKE3D will terminate. File *nwf* is complete, and may be used without modification for subsequent analysis.

3.3 Sense Switch Controls

Sense switch controls allow interactive interrogation of the progress of an analysis, the action of the nonlinear solver, and a means for controlled user intervention. NIKE3D sense switches and their actions are listed below. To activate a sense switch, first interrupt execution by entering `<ctrl-c>`, then enter the desired switch (including the terminating period, if any) at the prompt. On CRAY and SUN machines, simply enter the desired switch while NIKE3D is executing.

Type	Response
SW0.	NIKE3D terminates immediately
SW1.	A restart file is written and NIKE3D terminates
SW2.	NIKE3D responds with time, iteration, and convergence information
SW3.	NIKE3D responds every step with the time and iteration summary, and, if active, augmented Lagrangian convergence info is displayed (TOGGLE)
SW4.	A restart file is written and NIKE3D continues calculations.
SW5.	NIKE3D prompts for new timestep size
SW6.	NIKE3D prints SW2. information every iteration (TOGGLE).
SW7.	NIKE3D prints a continuous display of nonlinear solver progress (TOGGLE)
CONV	override convergence tolerances and proceed to next time step
PLOT	write plot state using current estimate of equilibrium, restart file, quit

Sense switches **SW1.**, **SW4.**, and **SW5.** must be processed after completion of a time step, so response to these switches may be delayed. TOGGLE-able sense switches may be enabled/disabled with repeated entry. Default initial settings are **SW3.** = `off`, **SW6.** = `off`, **SW7.** = `on`, and may be overridden in the input deck on Control Card 5. The **CONV** and **PLOT** switches allow the user to obtain non-equilibrium solutions for purposes of model debugging. These results must be interpreted with caution.

3.4 Runtime Errors

If an error occurs during a NIKE3D analysis, consult the high-speed printer file to determine NIKE3D's progress before the error occurred. A majority of errors are caused by improperly formatted or omitted data in the complex input deck. These errors can often be traced by comparing NIKE3D's echo of the input to the original deck.

Errors that occur after initialization often cause warning messages to be generated in the high-speed printer file. Most commonly these involve "loss of significant digits" or "change in sign" messages from the linear solver FISSLE. These messages provide the offending equation number. The global node number and degree of freedom associated with a particular equation number can be found in the Generated Nodal Data section of the high-speed printer file. Often by viewing the location of this node in the undeformed mesh, the input errors can be located. Users are cautioned that this type of error usually results from unconstrained rigid body modes in the model, which must be removed for static analyses.

The nonlinear equation solver presents another common source of difficulty. Many models pass the initialization phase of an analysis, but fail to converge in the equilibrium iteration phase. These problems are often due to modeling errors, and can be quite difficult to trace. In general, try to simplify the model enough to obtain a plot state to inspect visually with TAURUS. Using TAURUS, increase the displacement scale factor to magnify deformations in the model. Look for suspicious deformations, which may indicate faulty mesh construction or boundary condition specification. Some suggestions to obtain a plot state:

- Be sure that the plot interval is set to one.
- Activate sense switch SW6. to monitor convergence progress. The solution may be nearly converged when the iteration limit is reached. If necessary, temporarily relax the convergence tolerance (Control Card #6) or override using sense switch CONV.
- Temporarily decrease the time step or load increment. Infinitesimal loading may be necessary to yield a converged solution.
- Temporarily increase the time step or load increment. NIKE3D requires a numerically significant change in load each time step, and cannot find equilibrium if the solution at the end of a step is identical to that at the beginning of the step.
- Utilize sense switch PLOT to generate a plot state during the iteration process. This unconverged "result" will not reflect an accurate equilibrium solution, but may be helpful for model debugging.

4. INPUT FORMAT

This section describes the NIKE3D input file. The file is broken down into *decks* of logically associated *cards* (a terminology resurrected from the annals of computing when input files were keypunched onto cards, one line per card, which were then often banded together in decks.) Any card containing either of the characters * or \$ in column 1 is considered a comment card and ignored by NIKE3D. Alphanumeric data may appear in either upper or lower case. All input must appear in the sequential order defined below, i.e., control card 2 must follow control card 1, the material deck must follow the control deck, etc. Only the decks pertinent to the model need be defined.

4.1 Control Deck

The control deck, consisting of ten cards, is required for all models. Control information is used to determine which of the remaining decks are to be read. Each card contains one or more data fields, described by a column range and associated FORTRAN format descriptor. Asterisks preceding the column range of an entry indicate that an explanatory note is included in the control deck notes at the end of this subsection. Italicized variable names in parenthesis are assigned to many of the control card fields. These are referenced later in the input specifications, where each feature is described in greater detail.

Card 1		
Columns	Quantity	Format
*1-72	Heading to appear on output	12A6

Card 2

Columns	Quantity	Format
*1-2	Input format EQ. "FL": input follows this manual, NIKE3D v3.0 and later EQ. "EN": input follows 1991 manual, NIKE3D v2.0 and later EQ. "NN": input follows 1984 manual	A2
3-5	Number of materials	I3
6-15	Number of node points (<i>NUMNP</i>)	I10
16-25	Number of brick elements (<i>NUMELH</i>)	I10
26-35	Number of beam elements (<i>NUMELB</i>)	I10
36-45	Number of shell elements (<i>NUMELS</i>)	I10
46-50	Number of 1-D slidelines (<i>NUMID</i>)	I5
51-55	Number of sliding surfaces (<i>NUMSI</i>)	I5
56-60	Number of rigid walls and symmetry planes (<i>NRWSP</i>)	I5
61-65	Discrete element input flag (<i>INPDE</i>) EQ.0: no discrete elements EQ.1: discrete element deck specified	I5
66-70	Number of rigid nodes and facets (<i>NUMRNF</i>)	I5

Card 3

Columns	Quantity	Format
1-10	Number of time steps	I10
11-20	Time step size, dt	E10.0
*21-25	Auto time step control flag EQ. " ": use fixed time step size EQ. "AUTO": enable standard automatic step control EQ. "RATL": enable alternate automatic step control EQ. "COST": enable alternate automatic step control	1x, A4
*26-30	Maximum number of retries allowable per step (AUTO) EQ. 0: default = 5 (AUTO) Aggressiveness parameter, 1 a 10 (COST) EQ. 0: default = 2 (COST)	I5
31-35	Optimal number of iterations per step EQ. 0: default = 11	I5
36-45	Minimum allowable step size EQ. 0: default = $dt / 3$.	E10.0
*46-55	Maximum allowable step size LT. 0: load curve number for MUST POINT data EQ. 0: default = $3 * dt$	E10.0
56-60	Size of optimal iteration window (RATL only) EQ. n: step size unchanged if within n iterations of optimal value	I5

Card 4

Columns	Quantity	Format
1-5	Number of load curves (<i>NUMLC</i>)	I5
6-10	Maximum number of points defining any load curve	I5
11-15	Number of concentrated nodal loads (<i>NUMCNL</i>)	I5
16-20	Number of element surfaces having pressure loadings applied (<i>NUMPR</i>)	I5
21-25	Number of displacement boundary condition cards (<i>NUMDIS</i>)	I5
26-30	Number of beam elements with aerodynamic drag loads (<i>NUMADL</i>)	I5
31-35	Number of node constraint cards (<i>NRCC</i>)	I5
36-40	Body force loads due to base acceleration in x-direction (<i>BFBAX</i>) EQ.0: no x-acceleration NE.0: x-acceleration	I5
41-45	Body force loads due to base acceleration in y-direction (<i>BFBAY</i>) EQ.0: no y-acceleration NE.0: y-acceleration	I5
46-50	Body force loads due to base acceleration in z-direction (<i>BFBZ</i>) EQ.0: no z-acceleration NE.0: z-acceleration	I5
51-55	Body force loads due to angular velocity about x-axis (<i>BFAVX</i>) EQ.0: no x-angular velocity NE.0: x-angular velocity	I5
56-60	Body force loads due to angular velocity about y-axis (<i>BFAVY</i>) EQ.0: no y-angular velocity NE.0: y-angular velocity	I5
61-65	Body force loads due to angular velocity about z-axis (<i>BFAVZ</i>) EQ.0: no z-angular velocity NE.0: z-angular velocity	I5
66-70	Number of nodes with steering boundary conditions (<i>NSTEER</i>)	I5
71-75	Number of nodes with foundation boundary conditions (<i>NFNBC</i>)	I5

Card 5

Columns	Quantity	Format
1-5	Output printing interval GT.0: output is printed every n steps, excluding step 0. EQ.0: no printed output LT.0: output is printed every (-n) steps, including step 0.	I5
6-10	Output plotting interval EQ.0: plot data is generated every time step.	I5
*11-15	Number of node print blocks (<i>NNPB</i> 5) LT.0: no node point data printed EQ.0: all node points are printed	I5
16-20	Number of brick element print blocks (<i>NHPB</i> 5) LT.0: no brick element data printed EQ.0: all brick elements are printed	I5
21-25	Number of beam element print blocks (<i>NBPB</i> 5) LT.0: no beam element data printed EQ.0: all beam elements are printed	I5
26-30	Number of shell element print blocks (<i>NSPB</i> 5) LT.0: no shell element data printed EQ.0: all shell elements are printed	I5
*31-35	Number of time steps between running restart file generation EQ.0: no running restart file written EQ.n: running restart file overwritten every n steps	I5
*36-40	Number of time steps between restart file generation EQ.0: restart file written when NIKE3D terminates normally	I5
*41-45	Shell element strain dump flag EQ.1: include extra shell strain info in plot database	I5
*46-50	Initial sense switch toggles (up to five one-digit numbers) EQ.3: toggle default setting of SW3 . EQ.6: toggle default setting of SW6 . EQ.7: toggle default setting of SW7 .	5I1
*51-55	Acceleration data dump flag EQ.0: no acceleration data in plot database EQ.1: plot accelerations relative to base coordinate system EQ.2: plot absolute accelerations	I5
*56-60	Maximum augmented Lagrangian iterations per time step [MAXAUG]	I5
*61-65	Interface gap plot flag EQ.1: over-write y-velocity data with nodal gap data in plot database	I5

Card 6

Columns	Quantity	Format
*1-5	Nonlinear equilibrium solution method EQ.1: BFGS (required for augmented Lagrangians) (DEFAULT) EQ.2: Broyden EQ.3: Davidon-Fletcher-Powell EQ.4: Davidon symmetric EQ.5: modified Newton EQ.6: modified constant arc length EQ.7: modified constant arc length with line search EQ.8: modified constant arc length with BFGS EQ.9: modified constant arc length with Broyden EQ.10: modified constant arc length with DFP EQ.11: modified constant arc length with modified BFGS EQ.12: modified constant arc length with Davidon EQ.13: full Newton EQ.14: full Newton with line search	I5
*6-10	Bandwidth minimization flag EQ.0: bandwidth will not be minimized EQ.1: invoke new GPS bandwidth minimization EQ.2: invoke old GPS bandwidth minimization EQ.3: read nodal reorder data from input file	I5
*11-20	Number of steps between stiffness reformation at beginning of step EQ.0: default set to "1"	I10
*21-30	Number of time steps between equilibrium iterations EQ.0: default set to "1"	I10
*31-35	Maximum number of equilibrium iterations permitted between stiffness matrix reformations (<i>ILIMIT</i>) EQ.0: default set to "10"	I5
36-40	Maximum number of stiffness matrix reformations per time step EQ.0: default set to "15"	I5
*41-50	Convergence tolerance on displacements, <i>DTOL</i> LT.0.0: use new strategy, $DTOL = DTOL $ EQ.0.0: default set to "0.001" GT.0.0: use old strategy	E10.0
*51-60	Convergence tolerance on energy EQ.0.0: default set to "0.01"	E10.0
*61-70	Convergence tolerance on residual EQ.0.0: default set to "1.e+10" (deactivated)	E10.0
71-80	Convergence tolerance on line search EQ.0.0: default set to "0.90"	E10.0

Card 7

Columns	Quantity	Format
*1-5	Analysis type (<i>IMASS</i>) EQ.0: static analysis EQ.1: dynamic analysis EQ.2: dynamic analysis with stresses initialized statically at time = 0	I5
6-10	Initial condition parameter (<i>INTVEL</i>) EQ.0: initialize velocities to zero EQ.1: initial nodal velocities are read in	I5
*11-15	Thermal effects option (<i>ITEO</i>) EQ.0: no thermal effects EQ.N: nodal temperatures are given by load curve N, and are equal for all nodes in the mesh. EQ.-1: each time step, a new temperature state is read from a disk file. The time word at the beginning of each temperature state is ignored. EQ.-2: each time step, a temperature state is interpolated from the temperature states in a disk file. The time words at the beginning of each temperature state are used. EQ.-3: the disk file containing temperatures has only one state. The initial temperatures are assumed to be zero.	I5
*16-20	Temperature profile input flag (<i>ITPRO</i>) EQ. 0: no temperature cards given GT. 0: nodal temperature profile included in input	I5
*21-25	Number of eigenvalues and eigenvectors to be extracted (<i>NEIG</i>)	I5
*26-35	Frequency shift, cycles per unit time	E10.0
36-45	First Newmark integration parameter, EQ.0.0: default set to ".50"	E10.0
46-55	Second Newmark integration parameter, EQ.0.0: default set to ".25"	E10.0

Card 8

Columns	Quantity	Format
*1-5	Percent of memory option (<i>MEM</i>) EQ. 0: NIKE3D may use all available memory NE. 0: only <i>MEM</i> percent of available memory may be used (CRAY only) NE. 0: only <i>MEM</i> Mbytes of available memory may be used (non-CRAY)	I5
*6-15	Element data buffer size (CRAY => words, non-Cray => bytes) EQ. 0: NIKE3D computes value based on problem size EQ. n: max(NIKE3D's value, n)	I10
*16-20	Direct linear equation solver EQ. 1: FISSLE (in-core) (DEFAULT) EQ. 2: FISSLE (out-of-core) EQ. 3: ACTCOL (in-core) EQ. 4: PVS (vectorized in-core)	I5
*21-25	BFGS update vector storage option EQ. 1: store in core memory (DEFAULT) EQ. 2: store on disk	I5
*26-30	Brick element formulation EQ. 1: B-bar (DEFAULT) EQ. 2: B-bar with incompatible modes (note: add 10 to the above values for in-core storage of brick data)	I5
*31-35	Brick element geometric stiffness flag EQ. 1: neglect (DEFAULT) EQ. 2: include	I5
*36-40	Shell element formulation EQ. 1: Hughes-Liu (DEFAULT) EQ. 2: YASE EQ. 3: Belytschko-Tsay (DYNA3D-like) EQ. 4: Hughes-Liu with membrane thinning (note: add 10 to the above values for in-core storage of shell data)	I5
*41-50	Hourglass control parameter (Belytschko-Tsay shell only)	E10.0
*51-55	Shell element geometric stiffness flag EQ. 1: neglect (DEFAULT) EQ. 2: include	I5
*56-60	Beam element formulation EQ. 1: Hughes-Liu (DEFAULT) EQ. 11: Hughes-Liu with in-core storage of element data	I5
*61-65	Beam element geometric stiffness flag EQ. 1: neglect (DEFAULT) EQ. 2: include	I5

Card 9

Columns	Quantity	Format
*1-5	Number of unloading steps in modified arc length method (optional)	I5
*6-10	Solution method during arc length unloading EQ. 1: BFGS EQ. 2: Broyden EQ. 3: Davidon-Fletcher-Powell EQ. 4: Davidon symmetric EQ. 5: modified Newton	I5
*11-15	Node number for displacement controlled arc length method EQ. 0: generalized arc length method	I5
*16-20	Direction of displacement at arc length controlling node EQ. 1: global x-direction EQ. 2: global y-direction EQ. 3: global z-direction	I5
*21-30	Desired arc length EQ. 0: NIKE3D selects based on initial step size	E10.0
*31-35	Arc length constraint method EQ. 0: Crisfield EQ. 1: Ramm	I5
*36-40	Arc length damping option EQ. 0: no damping EQ. 1: damping applied	I5
41-45	Number of user-specified integration rules for beams (<i>NUSBIR</i>)	I5
46-50	Maximum number of user-specified integration points (beams)	I5
51-55	Number of user-specified integration rules for shells (<i>NUSSIR</i>)	I5
56-60	Maximum number of user-specified integration points (shells)	I5

Card 10

Columns	Quantity	Format
*1-5	Linear equation solver option EQ. 0: use direct linear equation solver chosen on Card 8 above EQ. 1: iterative solution with block diagonal scaling EQ. 2: iterative solution with Crout Element-By-Element EQ. 3: iterative solution with Gauss-Seidel EBE EQ. 4: iterative solution with Cholesky EBE	I5
*6-10	Iteration limit for linear solver EQ. 0: Default = 0.1*NUMNP	I5
*11-20	Iteration convergence tolerance EQ. 0: Default = 0.001	E10.0
21-25	Buffer size (elements) for out-of-core iterative linear solver (<i>LBUF</i>) EQ. 0: <i>LBUF</i> = 1024 LT. 0: double-buffered with <i>-LBUF</i> elements	I5
*26-30	Print-out option for linear iterative solver (default filename = <i>itrcont</i>) EQ. 0: no diagnostics printed to file EQ. 1: time step convergence information EQ. 2: iteration norm information EQ. 3: residual input, solution output EQ. 4: residual/solution each iteration	I5
*31-35	Iterative solver data storage option EQ. 1: store data in core memory (DEFAULT) EQ. 2: store data on disk	I5

Control Deck Notes

<u>(card) : (field)</u>	<u>Comments</u>
1 : 1	On 32-bit machines only the first 36 characters of the heading are used.
2 : 1	See the EXECUTION section for automatic conversion of old input decks to the new FL format.
3 : -	<p>Automatic Time Step Control Notes:</p> <p>If automatic time step control is not used, NIKE3D uses a constant step size equal to the input value, and terminates after the input number of steps. Failure of the equilibrium iteration process during a step results in error termination.</p> <p>If automatic time step control is selected, NIKE3D varies the step size according to the difficulty required to find equilibrium during each step. The termination time is computed by multiplying the input number of steps by the input time step size. The size of the final time step is adjusted to reach the termination time exactly. Failure of the equilibrium iteration process during a step results in a number of "retries" with progressively smaller time steps.</p>
3 : 3	<p>Fields 3:4 - 3:8 are ignored unless an auto time step control flag is enabled.</p> <p>The AUTO method computes a new time step size by applying a <i>constant multiplier</i> to the value used in the previous step. The iteration count from the previous step is compared to the optimum value to determine whether the step size should be increased or decreased. If a step fails, the step size is reduced linearly such that the minimum allowable value is reached on the last retry.</p> <p>The RATL method varies step size <i>exponentially</i> if the iteration count for a step falls outside the optimal iteration window.</p> <p>The COST method varies step size <i>exponentially</i>, keeping the degree of nonlinearity within each step constant. Thus a large number of iterations are allowed for steps which exhibit large nonlinearity. The minimum, maximum, and must point parameters are obeyed.</p>
3 : 4	Under AUTO time step control, this field indicates how many times NIKE3D will retry a failed step. Each retry uses progressively smaller steps, after which error termination results. Using the COST method, this parameter regulates the magnitude of step size changes.

Control Deck Notes (cont.)

<u>(card) : (field)</u>	<u>Comments</u>
3 : 7	<p>If a load curve is specified, the time values of each data point are times which NIKE3D must pass exactly. Function values at each data point indicate maximum allowable step size up to that time. Must point data should begin with the point (0,0). For example, to compute with a maximum time step size of 2.5 for three seconds, followed by a step size of 4.5 for one second, use the data pairs: (0,0) , (3.0,2.5) , (4.0,4.5). To obtain solutions at exactly one second intervals using the above maximum step size limits, use the data pairs: (0,0) , (1.0,2.5) , (2.0,2.5) , (3.0,2.5) , (4.0,4.5).</p> <p>By default, plot states are generated at each must point. To suppress plotting at a particular must point, enter a negative function value. The absolute value will be used to evaluate maximum allowable step size.</p>
5 : 3 - 5:6	Printout blocks provide a convenient way to limit often voluminous printed output to the sets of nodes and elements of particular interest.
5 : 7 - 5:8	Restart files may be used to resume execution of a partially completed analysis, or to extend a completed analysis. An optional restart input deck allows many parameters to be modified upon restart. Running restart files are overwritten, whereas regular restart files accumulate as the analysis proceeds. Check the screen or printfile for the names of restart files generated at various problem times.
5 : 9	This flag causes NIKE3D to (objectively) integrate shell strain data and include it in the plot database. This is required for accurate <i>finite</i> strain data.
5 : 10	The default start-up settings of sense switches sw3. , sw6. , and sw7. may be reversed by specifying the appropriate integer anywhere in the column range. For example, including a 36 or a 63 in the column range changes the default settings of sense switches sw3. and sw6. from off to on . Integers other than 3, 6, or 7 are ignored. See the EXECUTION section for more info on sense switches.
5 : 11	Nodal acceleration data computed from the Newmark equations may now be included in the TAURUS plot database. Previously, accelerations were obtained in TAURUS by differencing velocities. "Absolute" acceleration data includes any user-defined base acceleration boundary conditions.
5 : 12	If the augmentation loop remains unconverged after MAXAUG augmentations, NIKE3D accepts the solution and proceeds to the next step.
5 : 13	If the interface gap flag is activated, gap data computed by the sliding interface routines is included in the plot database. Gaps are computed between each interface <i>node</i> and its opposing <i>surface</i> . Negative values indicate penetration. Spatial contours and temporal variations of gap may then be displayed using the post-processor. Gap data is (temporarily) included in the database by over-writing the y-velocity data.

Control Deck Notes (cont.)

<u>(card) : (field)</u>	<u>Comments</u>
6 : 1	The BFGS solver is strongly recommended, and required when using the augmented Lagrangian contact and incompressibility constraints. By setting the number of iterations between reformations (<i>ILIMIT</i>) to one, the full Newton method can be emulated using the BFGS solver.
6 : 2	Since a bandwidth minimizer has been incorporated into NIKE3D, mesh nodes may be numbered in any convenient fashion. Because the minimizer does not renumber nodes, its use is transparent. The new GPS minimizer can accept arbitrarily large connectivity.
6 : 3 - 6 : 4	In nonlinear problems equilibrium iterations must always be permitted. Reformation of the stiffness matrix at the beginning of every time step often speeds convergence of the equilibrium search. Stiffness matrix reformation normally occurs once at the beginning of each step, and within a time step if either the equilibrium iteration limit is reached or the solution diverges.
6 : 3	To suppress the stiffness reformation at the beginning of each step, set this parameter to a value which exceeds the total number of steps in the problem. This is often economical for contact dominated problems, where the solution diverges immediately after the first equilibrium iteration, causing a (second) stiffness reformation.
6 : 4	To suppress equilibrium iterations during each step, set the parameter in this field to a value which exceeds the total number of time steps desired. A single linear solution will be performed each step. This solution may not represent an equilibrium state, hence results obtained may be inaccurate.
6 : 5	In problems dominated by contact it may be cost effective to limit the number of BFGS iterations to five or less between stiffness reformations. This allows NIKE3D to better account for rapidly changing contact areas. Enter a value of one to obtain Full Newton method with Line Search.
6 : 7	The old convergence strategy compares the current displacement increment to the total displacement <i>over the analysis</i> . This criterion becomes less strict as the analysis proceeds. The new strategy considers the total displacement <i>over the step</i> , hence is more consistent as the deformation evolves.
6 : 8	In problems where there is much rigid body motion, the displacement tolerance may be insufficient, and it may be advisable to tighten the energy tolerance to 10^{-3} .
6 : 9	The residual norm monitors the balance between internal and external forces in the model. By default, the residual tolerance is greater than one, effectively eliminating its influence on convergence.
7 : 1	This field is ignored for eigenvalue problems, i.e. when a nonzero number of eigenvalues and eigenvectors (<i>NEIG</i>) is requested.

Control Deck Notes (cont.)

<u>(card) : (field)</u>	<u>Comments</u>
7 : 3	<p>If <i>ITEO</i> is less than 0, nodal temperatures are read from the binary plot database generated by the heat conduction code, TOPAZ3D. This database is named on the execution line. If <i>ITEO</i> = -1, the number of states in TOPAZ3D's plotfile must equal or exceed the number of requested NIKE3D time steps. If <i>ITEO</i> = -2, and the NIKE3D problem time exceeds the maximum time defined in the TOPAZ3D database, the temperature profile from the last TOPAZ3D state is used. The temperature state at time zero is taken as the stress free state.</p> <p>For quasistatic thermal analyses (<i>IMASS</i> .eq. 0, <i>ITEO</i> .ne. 0) nodal temperatures are written to the NIKE3D hsp and plot files, and can be displayed using component 21 (labeled x-velocity) in TAURUS.</p>
7 : 4	<p>When <i>ITPRO</i> is greater than zero, a nodal temperature profile is input and used in conjunction with the thermal analysis option <i>ITEO</i> mentioned above. When <i>ITEO</i> is greater than zero, the thermal profile is evaluated at time t using the formula:</p> $\mathbf{T} = \mathbf{T}_{\text{mod}} * f(t) + \mathbf{T}_{\text{base}}$ <p>where \mathbf{T}_{mod} and \mathbf{T}_{base} are the profile input temperatures and $f(t)$ is the current value of load curve <i>ITEO</i>.</p> <p>When <i>ITEO</i> is less than 0, the value for $f(t)$ is given by load curve <i>ITPRO</i>. The above formula is used to evaluate the temperature profile <i>at time zero</i>. Temperatures for future times are taken from the TOPAZ input files according to <i>ITEO</i>.</p>
7 : 5	<p>If a frequency analysis is desired, NIKE3D will compute the desired number of eigenvalues and eigenvectors and will terminate. The normalized eigenvectors are written into the binary plotfile. TAURUS will plot the mode shapes. The circular frequency corresponding to the mode shape is printed in place of the time word on each plot.</p>
7 : 6	<p>If a frequency shift is specified, the <i>NEIG</i> eigenvalue/vector pairs nearest to that frequency will be determined, rather than the lowest <i>NEIG</i> pairs.</p>
8 : 1	<p>By allowing NIKE3D to consume more core memory, larger buffers for the element, constitutive, and stiffness data may be used. This reduces I/O cost at the potential expense of limiting cpu access on busy timeshared systems. Memory allocation on workstations is static, with a default maximum of 17.6 Mbytes. By requesting less than this amount, NIKE3D can be forced to utilize disk storage rather than core memory.</p>
8 : 2	<p>A larger element buffer reduces element I/O cost at the expense of limiting available core memory for stiffness matrix operations. See the "CONSTITUTIVE history data" printout at the beginning of each run for a summary of the size required by each element class. To optimize element loop speed, set the element buffer to the largest size indicated in the printout.</p>

Control Deck Notes (cont.)

<u>(card) : (field)</u>	<u>Comments</u>
8 : 3	<p>The stiffness matrix should be stored in-core if possible.</p> <p>The FISSLE solvers print valuable diagnostic messages to the high speed printer file when numerical errors are encountered.</p> <p>The ACTCOL solver often yields higher numerical precision on some workstations, especially SUNs.</p> <p>The PVS solver operates in vector mode on CRAY systems, and is very efficient on the IBM-RS/6000 machines. PVS often requires more storage than the other solvers (see Storaasli, Nguyen, and Agarwal [1990]).</p>
8 : 4	<p>The update vectors should be stored in-core if possible. For a problem with <i>NEQ</i> degrees of freedom and <i>ILIMIT</i> iterations permitted between stiffness matrix reformations, $2*NEQ*ILIMIT$ words of storage are required .</p>
8 : 5	<p>Element data should be stored in-core if possible. The incompatible modes formulation yields much improved accuracy for small deformation problems dominated by bending, at the expense of a modest increase in cpu and storage cost .</p>
8 : 6 , 8.9 , 8.11	<p>The geometric stiffness terms speed convergence for many problem classes, such as transversely loaded plates with clamped boundaries, where transverse deflections generate sharp increases in membrane stress. The geometric stiffness is required by hyperelastic material types #15 and #18.</p>
8 : 7	<p>Element data should be stored in-core if possible.</p> <p>The Hughes-Liu shell (Hughes, et.al. [1981b-d]) is a finite strain element derived from a degenerated eight node brick. Thinning effects due to finite membrane strain are optionally included, and reported in TAURUS as shell thickness component #69.</p> <p>The YASE shell (Engelmann, Whirley, and Goudreau [1989]) is a large deflection, small strain element with enhanced strain fields to accurately capture bending effects. It has proven to be more accurate than the Hughes-Liu shell for problems involving inplane shear and bending.</p> <p>The Belytschko-Tsay shell (Belytschko and Tsay [1984]) closely resembles the DYNA3D implementation, using one plan integration point. Developed to compare results with DYNA3D, it is not recommended for general use.</p>
8 : 8	<p>Hourglass control is required to suppress zero energy deformation modes which occur in the one point Belytschko-Tsay shell. See the DYNA3D manual (Whirley and Hallquist [1991]) for further details</p>
8 : 10	<p>Element data should be stored in-core if possible.</p>

Control Deck Notes (cont.)

<u>(card) : (field)</u>	<u>Comments</u>
9 : 1 - 9 : 7	The arc length solver is intended for use with buckling problems where the load-displacement curve is not monotonic, a problem class where traditional displacement or energy norm driven methods fail. In this method, the solution is advanced in increments of constant arc length on the load-displacement curve, regardless of its path. This may require load reduction or unloading as the problem passes a bifurcation point. Since documentation on NIKE3D arc length analyses is scarce, users are encouraged to relate their experiences.
9 : 3	The user may define a single node whose load-deflection history controls the solution. If no node is defined, norms of the global force and displacement vectors are used in the generalized arc length method.
9 : 4	Only displacement in the direction specified is used to calculate the controlling node's load-deflection curve.
9 : 5	The arc length is analogous to step size in traditional nonlinear solvers.
9 : 6	Crisfield's [1981] method uses a spherical constraint equation, while Ramm's [1980] uses a linearized method.
10 : 1	The element-by-element iterative linear equation solver dramatically reduces storage and cpu requirements for many problems. The block diagonal scaling method uses roughly half the storage of the other methods. The Crout, Gauss, and Cholesky preconditioned methods may converge faster. Selection is problem dependent. The iterative solver is not available for problems with beam or discrete elements, or rigid bodies.
10 : 2	Error termination occurs if a solution to the linear system is not obtained within the specified number of iterations.
10 : 3	Controls convergence of the iterative linear equation solver.
10 : 5	Printed information is recorded in file named on execution line with "C= " Option #2 prints convergence norms during iteration process.
10 : 6	Data should be stored in-core if possible.

4.2 Material Deck

Repeat the following set of cards for each material model. Most materials require an eight card set. Exceptions are noted. *Material card sets for use with beam, truss, shell, and membrane elements require two additional cross section definition cards.* These cards are described at the end of this section.

Card 1

Columns	Quantity	Format
1- 5	Material identification number	I5
6-10	Material type (letters in brackets identify element types for which the material is valid: <i>A</i> =brick, <i>B</i> =beam, <i>C</i> =H-L shell, <i>D</i> =YASE and B-T shell) EQ. 1: isotropic elastic [<i>A, B, C, D</i>] EQ. 2: orthotropic elastic [<i>A</i>] EQ. 3: elastic-plastic [<i>A, B, C, D</i>] EQ. 4: thermo-elastic-plastic [<i>A, C</i>] EQ. 5: soil and crushable foam [<i>A</i>] EQ. 6: viscoelastic [<i>A</i>] EQ. 7: thermo-orthotropic elastic [<i>A</i>] EQ. 8: thermo-elastic-creep [<i>A</i>] EQ. 9: power law elastic-plastic [<i>A</i>] EQ.10: power law thermo-elastic-plastic [<i>A</i>] EQ.11: transient thermal creep [<i>A</i>] EQ.12: ramberg-osgood elastoplasticity [<i>A</i>] EQ.13: general anisotropic thermo-elastic [<i>A</i>] EQ.14: oriented brittle damage [<i>A</i>] EQ.15: mooney-rivlin rubber [<i>A</i>] EQ.16: thermo-plastic melt [<i>A</i>] EQ.17: foundation boundary spring [none] EQ.18: transversely isotropic hyperelastic [<i>A</i>] EQ.19: strain rate sensitive power law plasticity [<i>C, D</i>] EQ.20: rigid body [<i>A, B, C, D</i>] EQ.23: thermo-orthotropic elastic laminate [<i>C</i>]	I5
*11-20	Density	E10.0
*21-25	Element class for which this material model is used EQ.1: brick elements EQ.2: beam or truss elements EQ.3: shell or membrane elements	I5
*26-35	Material reference temperature, T_m^{ref}	E10.0
*36-45	Rayleigh damping parameter	E10.0
*46-55	Rayleigh damping parameter	E10.0

Card 2

Columns	Quantity	Format
*1-72	Material title	12A6

Material Deck Notes - cards 1 and 2:

(card) : (field)	Comments
1 : 3	Mass density is used. For example, in English units, mass density has units of $\text{lb}\cdot\text{sec}^2\cdot\text{in}^{-4}$, where $1 \text{ lb}\cdot\text{sec}^2\cdot\text{in}^{-4} = 1.07\times 10^4 \text{ g/cc (approximately)}$
1 : 4	For element classes 1 and 2 (beam, truss, shell, and membrane elements), two cross section definition cards must be appended to the material card set. These are described at the end of this section.
1 : 5	The material reference temperature is required input for materials which have thermal expansion coefficients that vary with temperature (types 4, 8, 10, 11, 16). The difference of the reference and current temperatures is used to compute thermal strain. For materials with constant thermal expansion coefficients (types 7, 13), the reference temperature is taken as the temperature at time zero in the analysis.
1 : 6 - 1 : 7	The effects of proportional, or Rayleigh damping may be optionally included with any material type. The discrete damping matrix C is constructed from the discrete mass and elastic stiffness matrices M and K as $\mathbf{C} = \mathbf{M} + \mathbf{K}$. Damping is not included in eigenvalue analyses.
2 : 1	Only the first 36 characters are used in 32-bit machines.

Material Type 1 (Elastic)

Columns	Quantity		Format
1-10	Card 3	Young's modulus, E	E10.0
1-10	Card 4	Poisson's ratio,	E10.0
1-10	Card 5	Thermal expansion coefficient (beam element only)	E10.0
1-10	Card 6	Blank	
1-10	Card 7	Blank	
1-10	Card 8	Blank	

Linear material behavior is produced, where the rate of Cauchy stress $\dot{\sigma}$ evolves as a function of the rate of deformation $\dot{\epsilon}$ in the unrotated configuration according to:

$$\dot{\sigma} = (\text{tr } \dot{\epsilon}) \mathbf{I} + 2\mu \dot{\epsilon}$$

where λ and μ are the Lamé parameters:

$$\lambda = \frac{E}{(1+\nu)(1-2\nu)}$$

$$\mu = \frac{E}{2(1+\nu)}$$

For small strain linear analysis, other pertinent parameters are:

$$\text{Shear Modulus: } G = \frac{E}{2(1+\nu)} \quad \text{Bulk Modulus: } K = \frac{E}{3(1-2\nu)}$$

In this case the stress may be written directly as a function of strain \mathbf{e} :

$$\sigma = (\text{tr } \mathbf{e}) \mathbf{I} + 2\mu \mathbf{e}$$

The thermal expansion coefficient may be optionally included when Material Type 1 is used with beam elements.

Material Type 2 (Orthotropic Elastic)

Columns	Quantity		Format
1-30	Card 3	E_a, E_b, E_c (see Fig. 3)	3E10.0
1-30	Card 4	$\nu_{ab}, \nu_{ca}, \nu_{cb}$	3E10.0
1-30	Card 5	G_{ab}, G_{bc}, G_{ca}	3E10.0
1-10	Card 6	Material axes option, $AOPT$ EQ.0.0: locally orthotropic with material axes given by element nodes n_1, n_2 , and n_4 (see Fig. 4-1). Cards 7 and 8 below are blank. EQ.1.0: locally orthotropic with material axes determined by a point in space and global location of each element integration point. Card 8 below is blank. EQ.2.0: globally orthotropic with material axes determined using normalized vectors \mathbf{a} and \mathbf{d} defined below, where: $\mathbf{c} = \mathbf{a} \times \mathbf{d}$; $\mathbf{b} = \mathbf{c} \times \mathbf{a}$	E10.0
1-30	Card 7	x_p, y_p, z_p (defined for $AOPT = 1.0$)	3E10.0
1-30	Card 7	a_1, a_2, a_3 (defined for $AOPT = 2.0$)	3E10.0
1-30	Card 8	d_1, d_2, d_3 (defined for $AOPT = 2.0$)	3E10.0

For material type 2, the matrix that relates global stresses to global strains is defined as:

$$\mathbf{C} = \mathbf{T}^T \mathbf{C}_L \mathbf{T} \quad ,$$

where \mathbf{T} is a transformation matrix. \mathbf{C}_L is the local stiffness matrix, defined in terms of the Young's modulus E , shear modulus G , and Poisson's ratio along the orthogonal material directions a, b , and c . The local compliance matrix $\mathbf{S} = \mathbf{C}_L^{-1}$ has the simple form:

$$\mathbf{S} = \mathbf{C}_L^{-1} = \begin{array}{cccccc} \frac{1}{E_a} & \frac{-\nu_{ba}}{E_b} & \frac{-\nu_{ca}}{E_c} & 0 & 0 & 0 \\ \frac{-\nu_{ab}}{E_a} & \frac{1}{E_b} & \frac{-\nu_{cb}}{E_c} & 0 & 0 & 0 \\ \frac{-\nu_{ac}}{E_a} & \frac{-\nu_{bc}}{E_b} & \frac{1}{E_c} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{ab}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{bc}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{ca}} \end{array}$$

WARNING: NIKE3D requires $(\nu_{ba}, \nu_{ca}, \nu_{cb})$ as input, not the more conventional $(\nu_{ab}, \nu_{ac}, \nu_{bc})$.

For a uniaxial test conducted in the i -direction, Poisson's ratios are defined as $\nu_{ij} = -\frac{\epsilon_j}{\epsilon_i}$, where ϵ_j is a j -direction (off-axis) strain.

Symmetry of the compliance matrix \mathbf{S} gives the relations

$$\frac{\nu_{ba}}{E_b} = \frac{\nu_{ab}}{E_a}, \quad \frac{\nu_{ca}}{E_c} = \frac{\nu_{ac}}{E_a}, \quad \text{and} \quad \frac{\nu_{cb}}{E_c} = \frac{\nu_{bc}}{E_b}.$$

Furthermore, positive definiteness of the stiffness matrix \mathbf{C}_L requires that

$$\nu_{ba} < \sqrt{\frac{E_b}{E_a}}, \quad \nu_{ca} < \sqrt{\frac{E_c}{E_a}}, \quad \nu_{cb} < \sqrt{\frac{E_c}{E_b}}.$$

During initialization, NIKE3D generates a warning message if these conditions are not satisfied.

See Cook [1974] for further details.

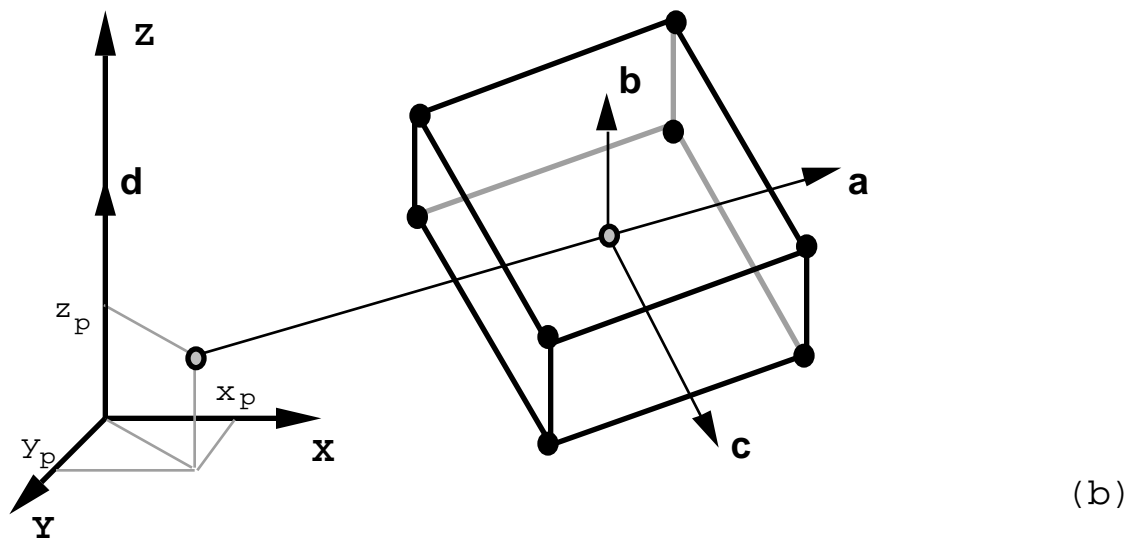
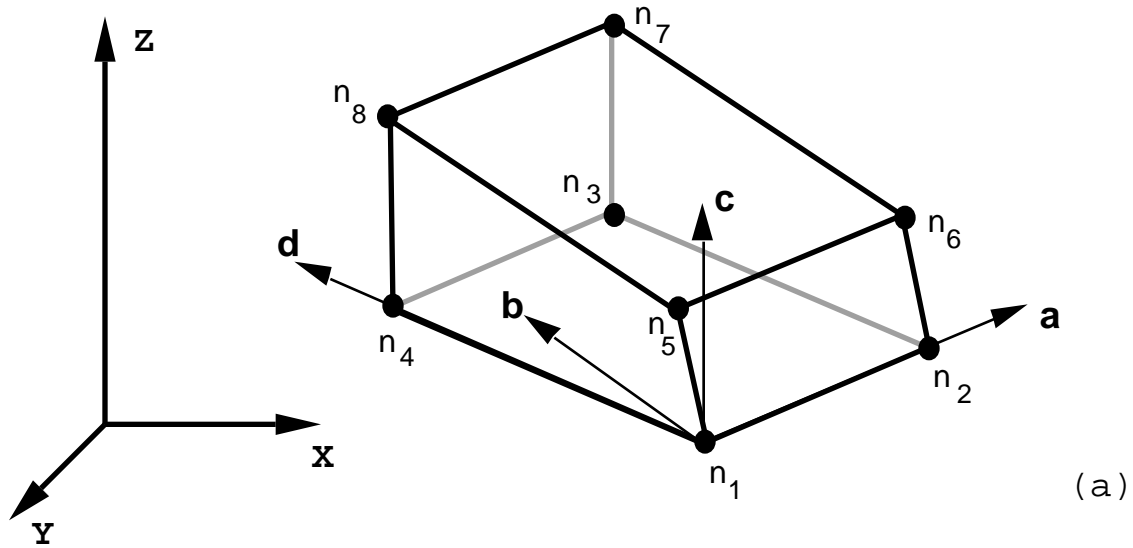


Figure 4-1. Options for determining principal material axes: (a) $AOPT = 0.0$, (b) $AOPT = 1.0$. Note that an orthonormal basis is formed from normalized vectors \mathbf{a} and \mathbf{d} using:

$$\mathbf{c} = \mathbf{a} \times \mathbf{d} \quad ; \quad \mathbf{b} = \mathbf{c} \times \mathbf{a}$$

Material Type 3 (Elastic-Plastic)

Columns	Quantity		Format
1-10	Card 3	Young's modulus	E10.0
1-10	Card 4	Poisson's ratio	E10.0
1-10	Card 5	Yield stress, σ_0	E10.0
1-10	Card 6	Hardening modulus	E10.0
11-20		Hardening parameter, $0 < \alpha < 1$ (See Fig. 4-2.)	E10.0
1-10	Card 7	$\bar{\epsilon}_1^p = 0.0$	E10.0
11-20		$\bar{\epsilon}_2^p$, effective plastic strain point 2	E10.0
. .		.	
. .		.	
. .		.	
71-80		$\bar{\epsilon}_8^p$ effective plastic strain point 8	E10.0
1-10	Card 8	$\bar{\sigma}_1$, effective stress at $\bar{\epsilon}_1^p$ (initial yield stress)	E10.0
11-20		$\bar{\sigma}_2$, effective stress at $\bar{\epsilon}_2^p$	E10.0
. .		.	
. .		.	
. .		.	
71-80		$\bar{\sigma}_8$, effective stress at $\bar{\epsilon}_8^p$	E10.0

This model represents plastic behavior, following Krieg and Key [1976]. Kinematic, isotropic, or a combination of kinematic and isotropic hardening may be specified by varying α between 0 and 1. For α equal to 0 and 1, respectively, kinematic and isotropic hardening are obtained as shown in Fig. 4-2. The measures of effective stress $\bar{\sigma}$, and strain $\bar{\epsilon}$, are defined below. If yield stress and hardening modulus (cards 5 and 6) are defined, cards 7 and 8 must be blank. An effective stress-effective plastic strain curve may be defined on cards 7 and 8. Effective stress is defined as

$$\bar{\sigma} = \sqrt{\frac{3}{2} s_{ij} s_{ij}},$$

where the deviatoric stress s_{ij} is given by:

$$s_{ij} = \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij},$$

and σ_{ij} is the Cauchy stress tensor. Effective plastic strain is given by

$$\epsilon^p = \int_0^t d\epsilon^p, \quad ,$$

where t denotes time and

$$d\epsilon^p = \frac{2}{3} d\epsilon_{ij}^p d\epsilon_{ij}^p \frac{1}{2} \quad .$$

When an effective stress - effective plastic strain curve (cards 7 and 8) is defined, cards 5 and 6 are ignored, and isotropic hardening is assumed. ϵ_1^p must be zero.

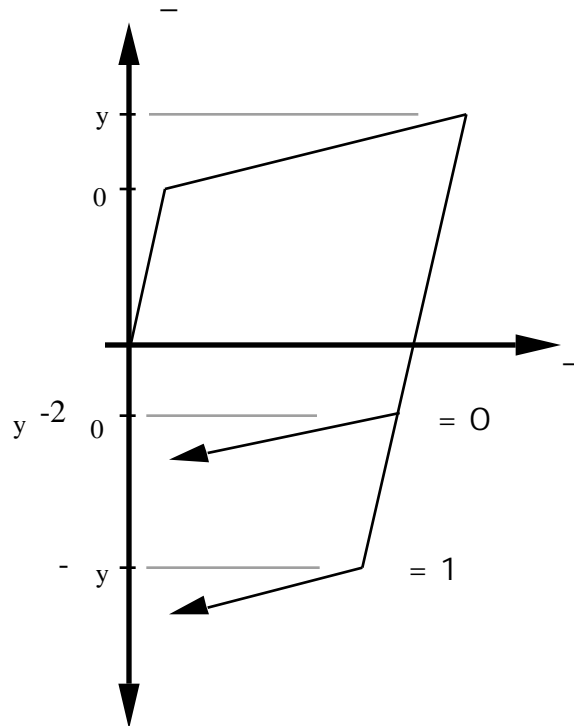


Figure 4-2. Elastic-plastic behavior during loading, followed by unloading and reverse loading in an idealized uniaxial test. The distinction between isotropic and kinematic hardening is evidenced by the increasing elastic range of the isotropic model.

Material Type 4 (Thermo-Elastic-Plastic)

Columns		Quantity	Format
1-10	Card 3	T_1 , temperature	E10.0
11-20		T_2 , temperature	E10.0
• •		•	
• •		•	
71-80		T_8 , temperature	E10.0
1-10	Card 4	E_1 , Young's modulus at T_1	E10.0
11-20		E_2 , Young's modulus at T_2	E10.0
• •		•	
• •		•	
71-80		E_8 , Young's modulus at T_8	E10.0
1-10	Card 5	ν_1 , Poisson's ratio at T_1	E10.0
11-20		ν_2 , Poisson's ratio at T_2	E10.0
• •		•	
• •		•	
71-80		ν_8 , Poisson's ratio at T_8	E10.0
1-10	Card 6	s_1 , secant coefficient of thermal expansion at T_1	E10.0
11-20		s_2 , secant coefficient of thermal expansion at T_2	E10.0
• •		•	
• •		•	
71-80		s_8 , secant coefficient of thermal expansion at T_8	E10.0
1-10	Card 7	y_1 , yield stress at T_1	E10.0
11-20		y_2 , yield stress at T_2	E10.0
• •		•	
• •		•	
71-80		y_8 , yield stress at T_8	E10.0
1-10	Card 8	E^p_1 , plastic hardening modulus at T_1	E10.0
11-20		E^p_2 , plastic hardening modulus at T_2	E10.0
• •		•	
• •		•	
71-80		E^p_8 , plastic hardening modulus at T_8	E10.0

At least two temperatures and their corresponding material properties must be defined. The analysis will be terminated if a material temperature falls outside the range defined in the input. If a thermo-elastic material is desired, leave cards 7 and 8 blank.

The *secant* coefficients of thermal expansion are defined with respect to the material reference temperature T_m^{ref} given on card 1 of the material deck. For problems where the coefficient of thermal expansion varies with temperature, the user must define secant (not tangent) coefficients. This distinction is depicted clearly in Fig. 4-3, where the variation of thermal strain is plotted with respect to temperature. Notice that in the case of thermally invariant expansion coefficients, the secant and tangent values coincide.

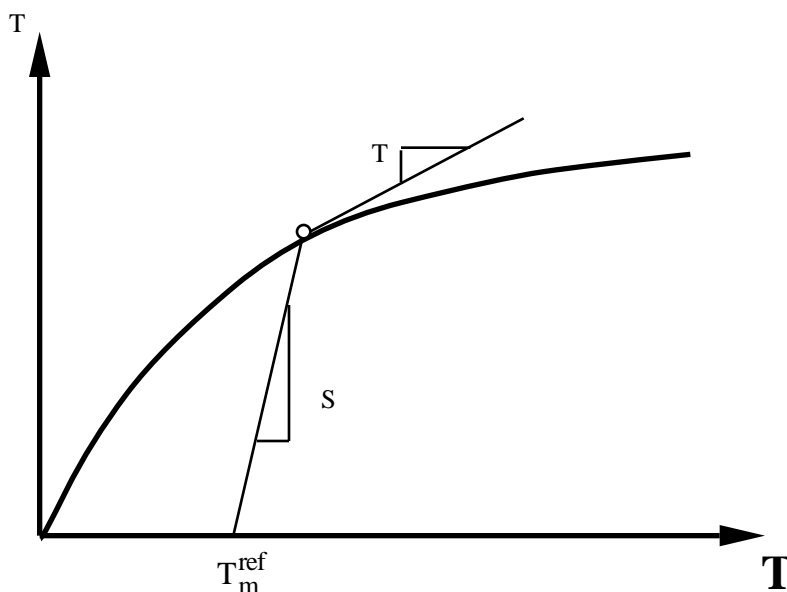


Figure 4-3 A hypothetical thermal strain vs. temperature curve is used to depict the difference between secant (S) and tangent (T) values of thermal expansion coefficients.

Material Type 5 (Soil and Crushable Foam)

Columns		Quantity	Format
1-10	Card 3	Shear modulus	E10.0
11-20		Bulk unloading modulus, K_u	E10.0
21-30		Yield function constant a_0	E10.0
31-40		Yield function constant a_1	E10.0
41-50		Yield function constant a_2	E10.0
51-60		Minimum pressure p_c (≤ 0.0) (see Fig. 4-4)	E10.0
61-70		Unloading option, UOPT EQ.0.0: volumetric crushing EQ.1.0: no volumetric crushing (see Fig. 4-4)	E10.0
1-40	Card 4	V_1, P_1, V_2, P_2 (Volumetric strain vs. Pressure data)	4E10.0
1-40	Card 5	V_3, P_3, V_4, P_4	4E10.0
1-40	Card 6	V_5, P_5, V_6, P_6	4E10.0
1-40	Card 7	V_7, P_7, V_8, P_8	4E10.0
1-40	Card 8	V_9, P_9, V_{10}, P_{10}	4E10.0

This model, originally developed by Krieg [1972], allows for the specification of a nonlinear pressure vs. volumetric strain relationship as shown in Fig. 4-4. Here, pressure is positive in compression, and volumetric strain is negative in compression. The tabulated data, specified two pairs per card on cards 4-8, must be in the order of increasing compression. Volumetric unloading behavior is illustrated in Fig. 4-4, including the optional effect of volumetric crushing. The unloading modulus in the (default) model with volumetric crushing is K_u . A minimum pressure p_c ($p_c < 0.0$) may also be defined. If pressure drops below the cutoff value, it is reset to the cutoff value and deviatoric stresses are zeroed.

The deviatoric behavior is elastic - perfectly plastic with the pressure dependent yield function given by:

$$= J_2 - (a_0 + a_1 p + a_2 p^2) \quad .$$

where

$$J_2 = \frac{1}{2} s_{ij} s_{ij} \quad ,$$

For nonzero values of a_1 or a_2 , the plastic flow is non-associative. On the yield surface, $J_2 = \frac{1}{3} \sigma_y^2$, where σ_y is the uniaxial yield stress (or equivalently the effective yield stress).

Therefore, the constants a_0 , a_1 , and a_2 may be determined from uniaxial test data and

$$\sigma_y = \left[3(a_0 + a_1 p + a_2 p^2) \right]^{\frac{1}{2}}$$

For elastic-perfectly plastic behavior $a_1 = a_2 = 0$, and $(3a_0)^{\frac{1}{2}}$ defines the yield strength.

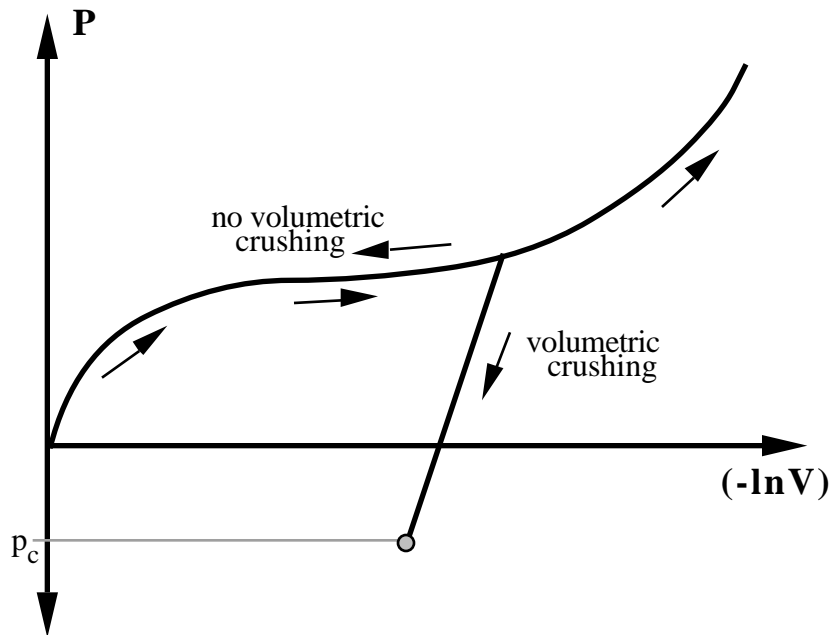


Figure 4-4. Pressure-volumetric strain curve for the soil and crushable foam model, where V is the relative volume.

Material Type 6 (Viscoelastic)

Columns	Quantity		Format
1-10	Card 3	Bulk modulus, K	E10.0
1-10	Card 4	Short time shear modulus, G_0	E10.0
1-10	Card 5	Long time shear modulus, G	E10.0
1-10	Card 6	Decay constant,	E10.0
	Card 7	Blank	
	Card 8	Blank	

In this model, described by Key [1974], the deviatoric stress rate is given by

$$\dot{s}_{ij} = 2 \int_0^t G(t - \tau) \dot{e}_{ij} d\tau$$

where the shear relaxation modulus $G(t)$ is described by

$$G(t) = G_\infty + (G_0 - G_\infty) e^{-t/\tau}$$

The volumetric response is elastic:

$$p = -K v$$

Material Type 7 (Thermo-Orthotropic Elastic)

Columns		Quantity	Format
1-10	Card 3	E_a Young's modulus in local a -direction (see Fig. 4-1)	E10.0
11-20		E_b Young's modulus in local b -direction	E10.0
21-30		E_c Young's modulus in local c -direction	E10.0
1-10	Card 4	ν_{ba} Poisson's ratio in local ba -direction (note: $\nu_{st} = \frac{-\epsilon_t}{\epsilon_s}$)	E10.0
11-20		ν_{ca} Poisson's ratio in local ca -direction	E10.0
21-30		ν_{cb} Poisson's ratio in local cb -direction	E10.0
31-40		α_a thermal expansion coefficient in local a -direction	E10.0
41-50		α_b thermal expansion coefficient in local b -direction	E10.0
51-60		α_c thermal expansion coefficient in local c -direction	E10.0
1-10	Card 5	G_{ab} shear modulus in local ab -direction	E10.0
11-20		G_{bc} shear modulus in local bc -direction	E10.0
21-30		G_{ca} shear modulus in local ca -direction	E10.0
1-10	Card 6	Material axes option, $AOPT$ EQ.0.0: locally orthotropic with material axes given by element nodes n_1 , n_2 , and n_4 (see Fig. 4-1). Cards 7 and 8 below are blank. EQ.1.0: locally orthotropic with material axes determined by a point in space and global location of each element integration point. Card 8 below is blank. EQ.2.0: globally orthotropic with material axes determined using normalized vectors \mathbf{a} and \mathbf{d} defined below, where: $\mathbf{c} = \mathbf{a} \times \mathbf{d}$; $\mathbf{b} = \mathbf{c} \times \mathbf{a}$	E10.0
1-30	Card 7	x_p, y_p, z_p (defined for $AOPT = 1.0$)	3E10.0
1-30	Card 7	a_1, a_2, a_3 (defined for $AOPT = 2.0$)	3E10.0
1-30	Card 8	d_1, d_2, d_3 (defined for $AOPT = 2.0$)	3E10.0

This model offers a thermally dependent extension of the orthotropic elastic model #2. For a description of the input coefficients, see model #2.

Material Type 8 (Thermo-Elastic-Creep)

Columns		Quantity	Format
1-10	Card 3	T_1 , temperature	E10.0
11-20		T_2 , temperature	E10.0
. .		.	
. .		.	
71-80		T_8 , temperature	E10.0
1-10	Card 4	$2G_1$, shear modulus at T_1	E10.0
11-20		$2G_2$, shear modulus at T_2	E10.0
. .		.	
. .		.	
71-80		$2G_8$, shear modulus at T_8	E10.0
1-10	Card 5	K_1 , bulk modulus T_1	E10.0
11-20		K_2 , bulk modulus T_2	E10.0
. .		.	
. .		.	
71-80		K_8 , bulk modulus at T_8	E10.0
1-10	Card 6	s_1 , secant coefficient of thermal expansion at T_1	E10.0
11-20		s_2 , secant coefficient of thermal expansion at T_2	E10.0
. .		.	
. .		.	
71-80		s_8 , secant coefficient of thermal expansion at T_8	E10.0
1-10	Card 7	a_{01}	E10.0
11-20		a_{02}	E10.0
. .		.	
. .		.	
71-80		a_{08}	E10.0
1-10	Card 8	b_{01}	E10.0
11-20		b_{02}	E10.0
. .		.	
. .		.	
71-80		b_{08}	E10.0

This model offers thermally dependent elastic creep behavior following Krieg [1977]. Here, G is the shear modulus, and the instantaneous creep rate is given by a power law of the form

$$\dot{\epsilon} = a_0 |\dot{\epsilon}|^{b_0}$$

At least two temperatures and their corresponding material properties must be defined. The analysis will be terminated if a material temperature falls outside the range defined in the input.

The *secant* coefficients of thermal expansion are defined with respect to the material reference temperature T_m^{ref} given on card 1 of the material deck. For problems where the coefficient of thermal expansion varies with temperature, the user must define secant (not tangent) coefficients. This distinction is depicted clearly in Fig. 4-3, where the variation of thermal strain is plotted with respect to temperature. Notice that in the case of thermally invariant expansion coefficients, the secant and tangent values coincide.

Material Type 9 (Power Law Plasticity)

Columns	Quantity		Format
1-10	Card 3	Young's modulus, E	E10.0
11-20		Poisson's ratio,	E10.0
21-30		Strength Coefficient k	E10.0
31-40		Hardening Exponent n	E10.0
Cards 4-8		Blank	

Elastic-plastic behavior with isotropic hardening is provided by this model. The effective yield stress $\bar{\sigma}_y$ increases as a function of initial yield strain ϵ_0 and effective plastic strain $\bar{\epsilon}^p$ according to the law:

$$\bar{\sigma}_y = k \left(\epsilon_0 + \bar{\epsilon}^p \right)^n$$

where the initial yield strain ϵ_0 is defined as

$$\epsilon_0 = \frac{E}{k} \frac{1}{n-1}$$

and effective quantities are computed using the standard norms (see material type 3).

Material Type 10 (Power Law Thermo-Elastic Plastic)

Columns		Quantity	Format
1-10	Card 3	T_1 , temperature	E10.0
11-20		T_2 , temperature	E10.0
. .		.	
. .		.	
. .		.	
71-80		T_8 , temperature	E10.0
1-10	Card 4	E_1 , Young's modulus at T_1	E10.0
11-20		E_2 , Young's modulus at T_2	E10.0
. .		.	
. .		.	
. .		.	
71-80		E_8 , Young's modulus at T_8	E10.0
1-10	Card 5	ν_1 , Poisson's ratio at T_1	E10.0
11-20		ν_2 , Poisson's ratio at T_2	E10.0
. .		.	
. .		.	
. .		.	
71-80		ν_8 , Poisson's ratio at T_8	E10.0
1-10	Card 6	α_{T_1} , tangent coefficient of thermal expansion at T_1	E10.0
11-20		α_{T_2} , tangent coefficient of thermal expansion at T_2	E10.0
. .		.	
. .		.	
. .		.	
71-80		α_{T_8} , tangent coefficient of thermal expansion at T_8	E10.0
1-10	Card 7	k_1 , strength coefficient at T_1	E10.0
11-20		k_2 , strength coefficient at T_2	E10.0
. .		.	
. .		.	
. .		.	
71-80		k_8 , strength coefficient at T_8	E10.0
1-10	Card 8	n_1 , hardening exponent at T_1	E10.0
11-20		n_2 , hardening exponent at T_2	E10.0
. .		.	
. .		.	
. .		.	
71-80		n_8 , hardening exponent at T_8	E10.0

Temperature sensitive elastic-plastic behavior is provided by this model, where the isothermal behavior is identical to that of material type 9. Generally, at least two temperatures and their corresponding material properties must be defined. The analysis will terminate if a material temperature falls outside the range defined in the input. If a thermo-elastic material is desired, leave cards 7 and 8 blank. If material constants are independent of temperature, input T_1 as zero on card 3, and define one set of material constants (or use material type 9).

Unlike other temperature-dependent material models, here the *tangent* coefficients of thermal expansion are defined *at* the given temperature. For problems where the coefficient of thermal expansion varies with temperature, the user must define tangent (not secant) coefficients. This distinction is depicted clearly in Fig. 4-3, where the variation of thermal strain is plotted with respect to temperature. Notice that in the case of thermally invariant expansion coefficients, the secant and tangent values coincide.

Material Type 11 (Transient Thermal Creep)

Columns	Quantity		Format
1-10	Card 3	Poisson's ratio	E10.0
11-20		Time exponent, m	E10.0
1-10	Card 4	T_1 , temperature	E10.0
11-20		T_2 , temperature	E10.0
· ·		·	
· ·		·	
· ·		·	
71-80		T_8 , temperature	E10.0
1-10	Card 5	E_1 , Young's modulus at T_1	E10.0
11-20		E_2 , Young's modulus at T_2	E10.0
· ·		·	
· ·		·	
· ·		·	
71-80		E_8 , Young's modulus at T_8	E10.0
1-10	Card 6	A_1 , Stress coefficient at T_1	E10.0
11-20		A_2 , Stress coefficient at T_2	E10.0
· ·		·	
· ·		·	
· ·		·	
71-80		A_8 , Stress coefficient at T_8	E10.0
1-10	Card 7	n_1 , Stress exponent at T_1	E10.0
11-20		n_2 , Stress exponent at T_2	E10.0
· ·		·	
· ·		·	
· ·		·	
71-80		n_8 , Stress exponent at T_8	E10.0
1-10	Card 8	s_1 , Secant coefficient of thermal expansion at T_1	E10.0
11-20		s_2 , Secant coefficient of thermal expansion at T_2	E10.0
· ·		·	
· ·		·	
· ·		·	
71-80		s_8 , Secant coefficient of thermal expansion at T_8	E10.0

By modifying the value of the time exponent m , the three classical creep regimes may be simulated:

$m < 1$	Primary creep
$m = 1$	Secondary (steady-state) creep. Use $m = 0.999$ for these calculations.
$m > 1$	Tertiary creep

In each case, the effective creep strain ϵ^c is given as:

$$\epsilon^c = A(T) t^{1/m(T)}$$

and may be plotted in TAURUS as component 7. See Whirley and Henshall [1990] for details of the formulation.

The *secant* coefficients of thermal expansion are defined with respect to the material reference temperature T_m^{ref} given on card 1 of the material deck. For problems where the coefficient of thermal expansion varies with temperature, the user must define secant (not tangent) coefficients. This distinction is depicted clearly in Fig. 4-3, where the variation of thermal strain is plotted with respect to temperature. Notice that in the case of thermally invariant expansion coefficients, the secant and tangent values coincide.

Material Type 12 (Ramberg-Osgood Elastic-Plastic)

Columns	Quantity		Format
1-10	Card 3	Reference shear strain, γ_y	E10.0
1-10	Card 4	Reference shear stress, τ_y	E10.0
1-10	Card 5	Stress coefficient,	E10.0
1-10	Card 6	Stress exponent, r	E10.0
1-10	Card 7	Bulk modulus, K	E10.0
	Card 8	BLANK	

The Ramberg-Osgood equation is an empirical constitutive relation to represent the one-dimensional elastic-plastic behavior of many materials. This implementation of the model allows a simple rate-independent representation of the hysteretic energy dissipation observed in materials subjected to cyclic shear deformation. The model is primarily intended as a simple model for shear behavior, as is often used in seismic analysis.

For monotonic loading, the stress-strain relationship is given by:

$$\begin{aligned} \frac{\tau}{\tau_y} &= \frac{\gamma}{\gamma_y} + \left| \frac{\gamma}{\gamma_y} \right|^r & \text{if } \dot{\gamma} > 0 \\ \frac{\tau}{\tau_y} &= \frac{\gamma}{\gamma_y} - \left| \frac{\gamma}{\gamma_y} \right|^r & \text{if } \dot{\gamma} < 0 \end{aligned}$$

where γ is the shear strain and τ is the shear stress. The model approaches perfect plasticity as the stress exponent $r \rightarrow \infty$. These equations must be augmented to correctly model unloading and reloading material behavior. The first load reversal is detected by $\dot{\gamma} < 0$. After the first reversal, the stress-strain relationship is modified to:

$$\frac{\tau - \tau_0}{2\tau_y} = \frac{\gamma - \gamma_0}{2\gamma_y} + \left| \frac{\gamma - \gamma_0}{2\gamma_y} \right|^r \quad \text{if } \dot{\gamma} > 0$$

$$\frac{\sigma_y}{2} = \frac{\sigma_y}{2} - \left| \frac{\sigma_y}{2} \right| \quad \text{if} \quad \dot{\sigma}_y < 0$$

where σ_y and $\dot{\sigma}_y$ represent the values of strain and stress at the point of load reversal. Subsequent load reversals are detected by $(\dot{\sigma}_y - \dot{\sigma}_y) < 0$.

The Ramberg-Osgood relations are inherently one-dimensional, and are assumed to apply to shear components. To generalize this theory to the multi-dimensional case, it is assumed that each component of the deviatoric stress and deviatoric tensorial strain is independently related by the one-dimensional stress-strain equations. A projection is used to map the result back into deviatoric stress space if required. The volumetric behavior is elastic, and therefore the pressure p is given by $p = -K_v \epsilon_v$, where ϵ_v is the volumetric strain.

The multi-dimensional extensions and implementation in NIKE3D are due to Drs. B.E. Engelmann and R.G. Whirley .

Material Type 13 (General Anisotropic Thermo-Elastic)

Columns		Quantity	Format
1-60	Card 3	$C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{16}$	6E10.0
1-60	Card 4	- $C_{22}, C_{23}, C_{24}, C_{25}, C_{26}$	10X,5E10.0
1-60	Card 5	- - $C_{33}, C_{34}, C_{35}, C_{36}$	20X,4E10.0
1-60	Card 6	- - - C_{44}, C_{45}, C_{46}	30X,3E10.0
1-60	Card 7	- - - - C_{55}, C_{56}	40X,2E10.0
1-60	Card 8	- - - - - C_{66}	50X,1E10.0
1-10	Card 9	a , Thermal expansion coefficient in local a-direction	E10.0
11-20		b , Thermal expansion coefficient in local b-direction	E10.0
21-30		c , Thermal expansion coefficient in local c-direction	E10.0
1-10	Card 10	Material axes option, <i>AOPT</i> EQ.0.0: locally orthotropic with material axes given by element nodes n_1 n_2 , and n_4 (see Fig. 4-1). Cards 11 and 12 below are blank. EQ.1.0: locally orthotropic with material axes determined by a point in space and global location of each element integration point. Card 12 below is blank. EQ.2.0: globally orthotropic with material axes determined using normalized vectors a and d defined below, where: $\mathbf{c} = \mathbf{a} \times \mathbf{d}$; $\mathbf{b} = \mathbf{c} \times \mathbf{a}$	E10.0
1-30	Card 11	x_p, y_p, z_p (defined for <i>AOPT</i> = 1.0)	3E10.0
1-30	Card 11	a_1, a_2, a_3 (defined for <i>AOPT</i> = 2.0)	3E10.0
1-30	Card 12	d_1, d_2, d_3 (defined for <i>AOPT</i> = 2.0)	3E10.0

Notice that this material card set contains twelve cards, not the standard eight. Cards 3-8 define the (symmetric) general anisotropic compliance matrix **C** in upper triangular form. Cards 10-12 define the local material axes using the convention of material type 2. See Fig. 4-1.

Material Type 14 (Oriented Brittle Damage)

Columns	Quantity		Format
1-10	Card 3	Young's modulus, E	E10.0
1-10	Card 4	Poisson's ratio,	E10.0
1-10	Card 5	Tensile strength, f_T	E10.0
11-20		Cracked shear strength, f_S	E10.0
21-30		Compressive yield strength, σ_y	E10.0
1-10	Card 6	Fracture toughness, g_c	E10.0
1-10	Card 7	Shear retention factor,	E10.0
1-10	Card 8	Viscosity,	E10.0

This model describes the anisotropic damage of brittle materials, and is designed primarily for application to concrete. The model admits progressive degradation of tensile and shear strengths across smeared cracks, which are initiated under tensile loadings. Damage, the evolution of cracks, is handled by treating the fourth rank elastic stiffness tensor as an evolving internal variable. Softening induced mesh dependencies are addressed by a characteristic length method described by Oliver [1989].

The elastic properties E and ν define the undamaged material response. When the first principal stress reaches the initial tensile strength, f_T , a smeared crack is initiated in the plane normal to the first principal stress direction. The crack orientation is fixed in the material, rotating in space with the body. As the loading advances, the tensile traction normal to the crack plane is progressively degraded to a small machine-dependent constant. This occurs by reducing the material's modulus normal to the smeared crack plane according to a maximum dissipation law that incorporates exponential softening. The normal traction $t_n = (\mathbf{n} \cdot \mathbf{n}) \cdot \boldsymbol{\sigma}$ is restricted by the condition: $t_n \leq f_T - (1 - \epsilon)f_T[1 - \exp(-H)]$, where \mathbf{n} is the smeared crack normal, ϵ is a small constant, $\boldsymbol{\sigma}$ is the stress, H is the softening modulus, and η is an internal variable. H is calculated automatically based upon the fracture toughness g_c and the element geometry.

The internal variable η , printed in the "effective plastic strain" field of the plot database, measures the crack field intensity. For plotting purposes, η is normalized such that $\eta \leq 5$. When the

value of α reaches one, the material's strength has been reduced to two percent of its original value in the directions normal and parallel to the smeared crack.

The shear traction t_s that can be transmitted across a smeared crack plane is similarly limited such that: $t_s = f_s(1 - \alpha)[1 - \exp(-H)]$, where f_s is the initial shear traction that may be transmitted and α is the shear retention factor. Shear degradation uses two orthogonal shear damage surfaces and is coupled to the tensile degradation through α . Shear degradation is achieved by reducing the material's shear stiffness parallel to the smeared crack plane. As the damage progresses, t_s approaches f_s .

The fracture toughness, g_c , is interpreted as the fracture energy per unit area of *crack advancement*.

Rate dependent or "viscous" behavior is implemented as a simple Perzyna regularization method, and is controlled by the material viscosity η . Choosing $\eta = 0$ gives rate-independent response.

Under compressive conditions, the material behavior is elastic / perfectly-plastic. A check on the stress state is performed using the J2 yield function $\sqrt{\frac{3}{2}}\|\mathbf{s} \cdot \mathbf{s}\| - \sigma_y \leq 0$, where \mathbf{s} is the stress deviator. If the yield function is violated, a J2 return mapping is executed and plastic strains are generated. If the stress state evolves such that a crack appears in a yielded material, plane stress plasticity is performed in the plane of the crack. If $\sigma_y = 0$, this plasticity is ignored. Currently there is no option to display the plastic strain values.

A variety of experimental data, from quasi-static to explosively driven situations, has been replicated using the model. An example of typical input parameters for a standard grade concrete, without rate effects, is:

$$\begin{aligned} E &= 3.15 \times 10^6 \text{ psi} & \sigma_y &= 3637 \text{ psi} \\ \alpha &= 0.2 & g_c &= 0.872 \text{ lbs/in} \\ f_{to} &= 450 \text{ psi} & \eta &= 0.03 \\ f_{so} &= 2100 \text{ psi} & \eta &= 0 \end{aligned}$$

A full description of the theoretical basis of the model is given by Govindjee, Kay, and Simo [1992]. Dr. Govindjee is responsible for the implementation in NIKE3D.

Material Type 15 (Mooney-Rivlin Rubber)

Columns	Quantity		Format
1-10	Card 3	Coefficient of first invariant term, A	E10.0
11-20		Coefficient of second invariant term, B	E10.0
21-30		Poisson's ratio,	E10.0
	Cards 4-7	Blank	
1-10	Card 8	Augmented Lagrangian flag	E10.0
		.EQ. 1.0: activate augmented Lagrangian method	
11-20		Convergence tolerance for augmented Lagrangian method	E10.0
		.GT. 0.0: converged when volume strain norm < <i>TOL</i>	
		.LT. 0.0: augment exactly (- <i>TOL</i>) times per step	

This hyperelastic material utilizes the so-called F-bar element formulation presented by Simo and Taylor [1991] to provide nearly- or fully-incompressible behavior while avoiding element locking. Incompressibility is achieved by penalty method, using the elastic bulk modulus as a penalty parameter. An optional augmented Lagrangian iteration loop enforces incompressibility to a user-specified tolerance. When using the augmented Lagrangian option, activate sense switch **SW3**. to obtain convergence information.

The strain energy density functional used is a Mooney-Rivlin type:

$$W = A(\tilde{I}_1 - 3) + B(\tilde{I}_2 - 3) + \frac{1}{2} K [\ln(\quad)]^2, \quad K = \frac{4(A+B)(1+\quad)}{3(1-2\quad)}$$

where coefficients A and B are entered above, and the K is a bulk modulus-like penalty parameter. The model differs from the standard Mooney-Rivlin form in that \tilde{I}_1 and \tilde{I}_2 are the invariants of the deviatoric part of the right Cauchy-Green tensor $\tilde{\mathbf{C}} = \tilde{\mathbf{F}}^T \tilde{\mathbf{F}}$, where $\tilde{\mathbf{F}} = J^{-1/3} \mathbf{F}$ is the deviatoric part of the deformation gradient and $J = \det[\mathbf{F}]$. In the incompressible limit, the values of J and the relative volume approach unity, and the standard Mooney-Rivlin form is recovered.

Material Type 16 (Thermo-Plastic Melt)

Columns	Quantity		Format
1-10	Card 3	Load curve for Young's modulus vs. T	E10.0
11-20		Load curve for Poisson's ratio vs. T	E10.0
21-30		Load curve for thermal expansion coefficient vs. T	E10.0
31-40		Flow Strength Model EQ. 0.0: elastic EQ. 1.0: rate independent elastic-plastic EQ. 2.0: rate dependent elastic-plastic EQ. 3.0: rate dependent elastic-plastic with recovery	E10.0
41-50		Fluid Flow Model EQ. 0.0: none EQ. 1.0: Newtonian	E10.0
51-60		Void Growth Model EQ. 0.0: none EQ. 1.0: pressure cut-off EQ. 2.0: Gurson-Tvergaard void growth	E10.0
61-70		Liquid - Solid Transition Model EQ. 0.0: none EQ. 1.0: fraction solid interpolation	E10.0
71-80		Variable Written To Plot Database EQ. 0.0: effective plastic strain EQ. 1.0: void strain EQ. 2.0: flow strength EQ. 3.0: effective strain rate	E10.0

Data for Flow Strength Model #1: Rate Independent, Elastic-Plastic

Columns	Quantity		Format
1-10	Card 4	Bulk Modulus (for sliding interface penalty)	E10.0
11-20		Load curve for Yield Stress vs. T	E10.0
21-30		Load curve for Plastic Tangent vs. T	E10.0

Data for Flow Strength Model #2: Rate Dependent, Elastic-Plastic

Columns		Quantity	Format
1-10	Card 4	Bulk Modulus (for sliding interface penalty)	E10.0
11-20		Load curve for Initial Strength, S_0 vs. T	E10.0
21-30		Load curve for Flow Strength Coefficient, A vs. T	E10.0
31-40		Load curve for Strain Rate Sensitivity, m vs. T	E10.0
41-50		Load curve for Strengthening Coefficient, B vs. T	E10.0
51-60		Load curve for Strengthening Exponent, n vs. T	E10.0

Data for Flow Strength Model #3: Rate Dependent, Elastic-Plastic with Recovery

Columns		Quantity	Format
1-10	Card 4	Bulk Modulus (for sliding interface penalty)	E10.0
11-20		Load curve for Initial Strength, S_0 vs. T	E10.0
21-30		Load curve for Flow Strength Coefficient, A vs. T	E10.0
31-40		Load curve for Strain Rate Sensitivity, m vs. T	E10.0
1-10	Card 5	Load curve for Strengthening Coefficient, k_1 vs. T	E10.0
11-20		Load curve for Strengthening Exponent, n_1 vs. T	E10.0
21-30		Load curve for Recovery Coefficient, k_2 vs. T	E10.0
31-40		Load curve for First Recovery Exponent, n_2 vs. T	E10.0
41-50		Load curve for Second Recovery Exponent, n_3 vs. T	E10.0

Data for Fluid Flow Model #1: Newtonian
--

Columns		Quantity	Format
1-10	Card 6	Load curve for Fluid Viscosity μ vs. T	E10.0

Data for Void Growth Model #1: Pressure Cut-Off
--

Columns		Quantity	Format
1-10	Card 7	Initial Void Strain	E10.0
11-20		Load curve for Tensile Pressure Cut-Off vs. T	E10.0
21-30		Load curve for Compressive Pressure Cut-Off vs. T	E10.0
31-70		Blank	40X
71-80		Consistent Elastic Moduli Flag (0 = off, 1 = on)	E10.0

Data for Void Growth Model #2: Gurson-Tvergaard
--

Columns		Quantity	Format
1-10	Card 7	Initial Void Strain	E10.0
11-20		First Gurson parameter, q_1	E10.0
21-30		Second Gurson parameter, q_2	E10.0
31-70		Blank	40X
71-80		Consistent Elastic Moduli Flag (0 = off, 1 = on)	E10.0

Data for Liquid-Solid Transition Model #1: Fraction Solid Interpolation
--

Columns		Quantity	Format
1-10	Card 8	Load curve for Fraction Solid f vs. T	E10.0

This material represents a collection of models intended for representing behavior of metals through the phase transition from liquid to solid, such as occurs during the solidification and cool-down phases of a casting operation. Phenomenological model types are selected on material card #3, and data appropriate for each model type is entered on subsequent cards. Cards associated with unused models may be left blank. Many of the parameters are specified through "load curve" numbers. These use NIKE3D's load curve feature to enter tabulated data specifying the parameter values versus temperature.

The first flow strength model, *rate independent elastic-plastic*, represents flow strength σ_f as a function of yield strength σ_y , effective plastic strain ϵ^p , and hardening modulus H :

$$\sigma_f = \sigma_y + H \epsilon^p$$

The second flow strength model, *rate dependent elastic-plastic*, represents flow strength as a function of strain rate $\dot{\epsilon}$, an evolving strength parameter S , and two material parameters A (Arrhenius) and m (strain sensitivity). S evolution is proportional to B and the plastic strain rate $\dot{\epsilon}^p$ according to:

$$\sigma_f = S \frac{\dot{\epsilon}^m}{A}, \quad \dot{S} = B (\dot{\epsilon}^p)^n$$

The third flow strength model, *rate dependent elastic-plastic with recovery*, is a variant of the second model where the strength coefficient is formed from a strength constant term S^o , and an evolving strengthening coefficient S_H . Evolution of S_H is controlled by hardening and dynamic relaxation terms according to:

$$S = S^o + S_H, \quad \dot{S}_H = k_1 (\dot{\epsilon}^p)^{n_1} - k_2 (\dot{\epsilon}^p)^{n_2} (S_H)^{n_3}$$

The fluid flow model is a Newtonian law: $\sigma = \mu \dot{\epsilon}$. For dual phase conditions, the continuum stress is determined from a weighted average of fluid and solid flow stresses, where the weight factors are taken from the fraction solid curve, evaluated at the current temperature.

The pressure cut-off void growth model is a phenomenological mechanism to prevent unrealistic hydrostatic stress states in the liquid regime. Voids are generated when tensile cut-off occurs. Void strains decrease when compressive cut-off is reached. The Gurson-Tvergaard void growth model is a solid phase model for void growth evolution. This model alters the yield function by including a void fraction f , and two additional parameters q_1 and q_2 :

$$\frac{\sigma}{\sigma_f} = \frac{1}{2} + 2f q_1 \cosh \frac{k k}{2 f} - (1 + q_2 f^2), \quad \dot{f} = (1 - f) \dot{\epsilon}^p_{kk}$$

A full description of the thermo-plastic melt model is given by Raboin [1994]. Dr. Raboin is responsible for the implementation in NIKE3D.

Material Type 17 (Foundation Boundary Spring)

Columns	Quantity							Format
1-60	Card 3	K_{11}	K_{12}	K_{13}	K_{14}	K_{15}	K_{16}	6E10.0
1-60	Card 4	-	K_{22}	K_{23}	K_{24}	K_{25}	K_{26}	10X,5E10.0
1-60	Card 5	-	-	K_{33}	K_{34}	K_{35}	K_{36}	20X,4E10.0
1-60	Card 6	-	-	-	K_{44}	K_{45}	K_{46}	30X,3E10.0
1-60	Card 7	-	-	-	-	K_{55}	K_{56}	40X,2E10.0
1-60	Card 8	-	-	-	-	-	K_{66}	50X,1E10.0

This "material" is used to input the stiffness matrix associated with the foundation node boundary condition. The upper triangle of the symmetric stiffness matrix \mathbf{K} is entered above. Global foundation node reaction forces f_i and moments m_i are computed from global foundation node displacements u_i and rotations θ_i as $\mathbf{F} = \mathbf{K}\mathbf{x}$, where

$$\mathbf{F} = \begin{bmatrix} f_x & f_y & f_z & m_x & m_y & m_z \end{bmatrix}^T \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} u_x & u_y & u_z & \theta_x & \theta_y & \theta_z \end{bmatrix}^T.$$

The stiffness matrix need not be full rank - one or more rows and/or columns may be zero to allow unconstrained motion in selected global directions.

Material Type 18 (Transversely Isotropic Visco-Hyperelasticity)

Columns		Quantity	Format
1-10	Card 3	Mooney-Rivlin Coefficient, C_1	E10.0
11-20		Mooney-Rivlin Coefficient, C_2	E10.0
21-30		Exponential Stress Coefficient, C_3	E10.0
31-40		Fiber Uncrimping Coefficient, C_4	E10.0
41-50		Modulus of Straightened Fibers, C_5	E10.0
1-10	Card 4	Bulk Modulus, K	E10.0
11-20		Fiber stretch for straightened fibers, *	E10.0
1-10	Card 5	Initial Stretch Flag .EQ. 1.0: activate initial stretch option	E10.0
11-20		Load Curve, n , for applying initial stretch	E10.0
21-30		Augmented Lagrangian Flag .EQ. 1.0: activate augmented Lagrangian method	E10.0
31-40		Tolerance for augmented Lagrangian iterations, TOL .GT. 0.0: converged when volume strain norm $< TOL$.LT. 0.0: augment exactly ($-TOL$) times per step	E10.0
1-10	Card 6	Material axes option, $AOPT$ Note: Fiber direction is always aligned with local axis a . EQ.0.0: local material axes given by local element nodes specified on Card 7 below. Fig 4-1 indicates local axes resulting from default node choices. Card 8 is blank. EQ.1.0: local material axes determined by a point in space and global location of each element integration point (see Fig. 4-1). Card 8 below is blank. EQ.2.0: local material axes determined using normalized vectors a and d defined below, where: c = a x d ; b = c x a	E10.0
1-30	Card 7	AOPT.EQ.0.0: local element nodes (default = 1,2,4)	3E10.0
1-30		AOPT.EQ.1.0: x_p, y_p, z_p	3E10.0
1-30		AOPT.EQ.2.0: a_1, a_2, a_3	3E10.0
1-30	Card 8	d_1, d_2, d_3 (used only for $AOPT = 2.0$)	3E10.0

This material provides a capability for modeling fiber reinforced visco-hyperelastic materials. Large strain hyperelastic response is determined by the strain energy formulation described in Section 1. Rate dependence can be added to the model by prescribing relaxation times as described in Section 2.

Section 1.: Hyperelastic Response

The isotropic behavior of the Mooney Rivlin Rubber model (material #15) is modified by the addition of a strain energy term $F(\cdot)$ resulting from stretch λ along a fiber direction:

$$W = C_1(\tilde{I}_1 - 3) + C_2(\tilde{I}_2 - 3) + \frac{1}{2}K[\ln(\lambda)]^2 + F(\lambda)$$

The fiber stretch λ along the current fiber direction \mathbf{a} is given by $\mathbf{a} = \mathbf{F}\mathbf{a}_0$, where \mathbf{F} is the deformation gradient, and \mathbf{a}_0 is the fiber direction in the undeformed configuration. The fiber strain term $F(\lambda)$ includes an exponential "toe region", followed by a linear region, to represent the uncrimping behavior of fibers in biological soft tissues. The transition from exponential to linear behavior is governed by specifying a fiber stretch λ^* . The material model was designed to represent the behavior of ligaments and tendons, however the exponential behavior of skeletal and cardiac muscle may be represented by setting λ^* to a large value.

The additional contribution to Cauchy stress due to the fiber stretch is given by the term:

$$\frac{F}{\lambda} = \frac{C_3}{\lambda} \left(\exp(C_4(\lambda - 1)) - 1 \right), \quad \lambda < \lambda^*$$

$$\frac{1}{\lambda} (C_5 + C_6), \quad \lambda \geq \lambda^*$$

Here, C_3 scales the exponential stresses, C_4 controls the rate of uncrimping of the collagen fibers, C_5 is the modulus of the straightened fibers, and C_6 is chosen by NIKE3D to insure C^0 continuity between the two functional regions.

The model also supports the application of initial tension to the material through specification of an initial fiber stretch, λ_0 . The initial stretch is applied using load curve n , specified above.

Details of the elastic constitutive model and its implementation can be found in Weiss [94]. Dr. Weiss is responsible for the NIKE3D implementation of this model.

Section 2.: Viscoelastic Response

The following supplemental input provides rate dependence to the material model described in Section 1. If columns 51-60 on Card 3 are blank, the material defaults to the non-rate dependent hyperelastic behavior.

Columns		Quantity	Format
51-60	Card 3	viscoelastic coefficient ₁	E10.0
61-70		viscoelastic coefficient ₂	E10.0
71-80		viscoelastic coefficient ₃	E10.0
51-60	Card 4	relaxation time ₁	E10.0
61-70		relaxation time ₂	E10.0
71-80		relaxation time ₃	E10.0
51-60	Card 5	viscoelastic coefficient ₄	E10.0
61-70		viscoelastic coefficient ₅	E10.0
71-80		viscoelastic coefficient ₆	E10.0
51-60	Card 6	relaxation time ₄	E10.0
61-70		relaxation time ₅	E10.0
71-80		relaxation time ₆	E10.0

Viscoelasticity is provided by adding additional time dependent stress terms to the elastic stress calculated above. The following formula is used to calculate the 2nd Piola Kirchoff stress at time t .

$$S(E, t) = \frac{W}{E}(E(t)) + \int_0^t G(t - \tau) \frac{W}{E}(E(\tau)) d\tau$$

where W is the strain energy as given above and E is the Green strain. The equilibrium elastic response is given by the first term on the right hand side of the above formula. A six term Prony series is used for the kernel $G(t)$ such that

$$G(t) = \sum_{i=1}^6 g_i \exp(-t/\tau_i)$$

Up to six viscoelastic coefficients g_i and relaxation times τ_i can be prescribed as shown in the table above for the material model. Different methods can be used to fit the viscoelastic parameters to experimental data. One such method is given in Puso and Weiss [96].

Material Type 19 (Strain Rate Sensitive Power Law Plasticity)

Columns	Quantity		Format
1-10	Card 3	Young's modulus, E	E10.0
11-20		Poisson's ratio,	E10.0
21-30		Strength Coefficient k .	E10.0
31-40		Hardening Exponent n	E10.0
41-50		Strain rate sensitivity exponent m	E10.0
51-60		Initial strain rate (DEFAULT=0.0002)	E10.0
Cards 4-8		Blank	

Elastic-plastic behavior with isotropic hardening is provided by this model. If any of the parameters k , n , or m are negative, their absolute value identifies a load curve giving their variation with respect to the effective plastic strain $\bar{\epsilon}^p$ for k and n and effective strain rate $\dot{\bar{\epsilon}}$ for m . The effective yield stress $\bar{\sigma}_y$ increases as a function of initial yield strain $\bar{\epsilon}_0$ and $\bar{\epsilon}^p$ according to the law:

$$\bar{\sigma}_y = k \dot{\bar{\epsilon}}^m \left(\bar{\epsilon}_0 + \bar{\epsilon}^p \right)^n$$

where $\dot{\bar{\epsilon}}$ is the effective strain rate, the initial yield strain $\bar{\epsilon}_0$ is defined as

$$\bar{\epsilon}_0 = \frac{E}{k \dot{\bar{\epsilon}}^m} \frac{1}{n-1}$$

and effective quantities are computed using the standard norms (see material type 3).

Material Type 20 (Rigid Body)

Columns	Quantity		Format
1-10	Card 3	Young's modulus, E	E10.0
1-10	Card 4	Poisson's ratio,	E10.0
1-10	Card 5	X-trans. boundary condition code	E10.0
11-20		Y-trans. boundary condition code	E10.0
21-30		Z-trans. boundary condition code	E10.0
31-40		X-rot. boundary condition code	E10.0
41-50		Y-rot. boundary condition code	E10.0
51-60		Z-rot. boundary condition code	E10.0
1-10	Card 6	Center of mass input flag NE. 1.0: NIKE3D computes center of mass EQ. 1.0: read user-supplied coordinates below	E10.0
11-20		X-coordinate of center of mass	E10.0
21-30		Y-coordinate of center of mass	E10.0
31-40		Z-coordinate of center of mass	E10.0

Cards 7-8 Blank

Rigid bodies are processed internally using a rigid body mechanics approach, rather than as "stiff" finite elements. Thus rigid elements are skipped during NIKE3D's normal element processing. Instead, surface tractions and boundary conditions are resolved onto the center of mass coordinates. Motion of the rigid body is determined by solving a coupled set of equilibrium equations for the rigid body and any other deformable elements in the model. Then nodal positions are updated using the motion of the center of mass and the rigid body assumption.

Rigid bodies may be constructed using brick, beam, and/or shell elements. Rigid bodies with unique material numbers that share common node points are automatically merged, and treated as a single rigid body. Each rigid body has at most six global degrees of freedom. All rigid body node points have six local degrees of freedom, regardless of element type. Thus if an elastic beam is connected at a single node to a rigid body constructed from brick elements, the joint is flexurally and torsionally rigid (not pinned, as would be the case if the brick elements were also elastic).

Elastic properties are used for sizing penalty parameters used in contact interfaces. Reasonable values should be entered.

Nodal constraints and displacement boundary conditions on rigid body *nodes* are ignored. Instead, constraints are applied at the rigid body center of mass. The center of mass is either computed by NIKE3D, or input by the user. The boundary condition codes are interpreted as:

<i>code</i> .lt. 0 :	fixed
<i>code</i> .eq. 0 :	free
<i>code</i> .gt. 0 :	number of load curve for prescribed displacement or rotation (radians)

Reaction forces at prescribed degrees of freedom are included in the printed output. To obtain reaction forces for fixed degrees of freedom, use a prescribed boundary condition with a zero load curve value.

Rigid bodies are currently available for static analyses only. Material density is required, and is used to compute center of mass coordinates and body forces due to base accelerations. Details of the formulation are presented in Maker [1995].

Material Type 23 (Thermo-Orthotropic Elastic Laminate)

Columns	Quantity		Format
1-10	Card 3	Number of points in material constant vs. temperature curves ($1 < \text{NUMPTS} < 49$)	E10.0
1-80	Card 4	Blank	80X
1-80	Card 5	Blank	80X
1-10	Card 6	Material axes option, <i>AOPT</i> EQ.0.0: locally orthotropic with material axes given by element nodes n_1 , n_2 , and n_4 (see Fig. 4-1). Cards 7 and 8 below are blank. EQ.1.0: locally orthotropic with material axes determined by a point in space and global location of each element integration point. Card 8 below is blank. EQ.2.0: globally orthotropic with material axes determined using normalized vectors a and d defined below, where: $\mathbf{c} = \mathbf{a} \times \mathbf{d}$; $\mathbf{b} = \mathbf{c} \times \mathbf{a}$ EQ.3.0: applicable to shell elements only. This option determines locally orthotropic material axes by offsetting the material axes by an angle (Card 8) from a line in the plane of the shell determined by taking the cross product of the vector v defined on Card 7 with the shell normal vector.	E10.0
1-30	Card 7	x_p, y_p, z_p (defined for $AOPT = 1.0$)	3E10.0
1-30	Card 7	a_1, a_2, a_3 (defined for $AOPT = 2.0$)	3E10.0
1-30	Card 7	v_1, v_2, v_3 (defined for $AOPT = 3.0$)	3E10.0
1-30	Card 8	d_1, d_2, d_3 (defined for $AOPT = 2.0$)	3E10.0

(Insert here the shell cross section definition cards 9 and 10, described at the end of this section)

Continue the material 23 deck by defining the following card sets for each of the 12 orthotropic constants, followed by the list of corresponding temperatures, using the format (8E10.0). This data must follow cards 9 and 10, the shell cross section definition cards.

Columns	Quantity	Format
1-10	E_a at temperature T_1	E10.0
11-20	E_a at temperature T_2	E10.0
•	•	•
•	•	•
•	•	•
71-80	E_a at temperature T_8	E10.0

Continue on additional cards until NUMPTS points have been defined.

Continue by defining card sets for the remaining eleven orthotropic constants and temperature, in the order: E_b , E_c , ν_{ba} , ν_{ca} , ν_{cb} , ν_{ab} , ν_{bc} , ν_{ca} , G_{ab} , G_{bc} , G_{ca} , T (the list of temperatures). For each card set, use the format described above for E_a .

On the next card(s), define the material angle for each through the thickness integration point:

Columns	Quantity	Format
1-10	1 material angle at first integration point	E10.0
11-20	2 material angle at second integration point	E10.0
•	•	•
•	•	•
•	•	•
71-80	8 material angle at eighth integration point	E10.0

Continue on additional cards as necessary until NIP points have been defined, where NIP is the number of thickness integration points specified on Card 9.

Cross Section Definitions - Beam and Truss Materials

The following two cards are added to the material deck for beam and truss materials only.

Card 9

Columns	Quantity	Format
1-10	Shear area factor EQ.0.0: default set to 5/6	E10.0
11-20	Quadrature rule for beam cross section LT.0.0: absolute value gives user-defined rule number EQ.0.0: default set to 2.0 EQ.1.0: truss element EQ.2.0: beam element with 2x2 Gauss quadrature in cross section EQ.3.0: beam element with 3x3 Gauss quadrature in cross section EQ.4.0: beam element with 3x3 Lobatto integration in cross section EQ.5.0: beam element with 4 x 4 Gauss quadrature in cross section	E10.0
21-30	Cross section shape EQ.0.0: rectangular EQ.1.0: tubular EQ.2.0: user-defined integration rule	E10.0

The default shear area factor is appropriate for many rectangular cross section beam applications. For solid circular cross sections, a value of 0.9 is more appropriate. For thin-walled tubular cross sections, use 2.0. See the GEMINI manual (Murray [1984]) or an elasticity text for further details.

Truss elements have no torsional or bending stiffness. Rotational degrees of freedom at truss nodes must be constrained, either by boundary conditions or the presence of a beam or shell element.

Card 10

Columns	Quantity	Format
1-10	Cross section dimension d_{s1} at node n_1 rectangular: d_{s1} = width in s direction (see fig. 4-6) tubular: d_{s1} = outside diameter	E10.0
11-20	Cross section dimension d_{s2} at node n_2 rectangular: d_{s2} = width in s direction tubular: d_{s2} = outside diameter	E10.0
21-30	Cross section dimension d_{t1} at node n_1 rectangular: d_{t1} = width in t direction tubular: d_{t1} = inside diameter (zero for solid bar)	E10.0
31-40	Cross section dimension d_{t2} at node n_2 rectangular: d_{t2} = width in t direction tubular: d_{t2} = inside diameter (zero for solid bar)	E10.0
41-50	Location of reference surface normal to s -axis (rectangular only) EQ. 1.0: side at $s = 1$ EQ. 0.0: center EQ. -1.0: side at $s = -1.0$	E10.0
51-60	Location of reference surface normal to t -axis (rectangular only) EQ. 1.0: side at $t = 1$ EQ. 0.0: center EQ. -1.0: side at $t = -1.0$	E10.0

Cross section dimensions specified above are overridden by any nonzero values specified on the element cards. For tubular cross section beams, dimensions in the s and t directions represent outer and inner diameters, respectively. Solid circular cross sections may be obtained by setting the t dimension to zero.

The reference surface position defines the local origin within the cross section. This is used in computations of resultant forces and moments. This location can be crucial when beams, shells, and solids are interconnected. For example, a beam element used as a stringer reinforcement to a shell surface may be made symmetric (extending equally from upper and lower shell surfaces) or nonsymmetric (extending outward from only one surface) by changing the location of the reference surfaces. The reference surface must be located at the center of tubular cross sections.

Cross Section Definitions - Membrane and Shell Materials

The following two cards are added to the material deck for shell and membrane materials only.

Card 9

Columns	Quantity	Format
1-10	Shear factor (default = 1.0)	E10.0
11-20	Number of through shell thickness integration points EQ.1.0: 1 point (membrane element) EQ.2.0: 2 points EQ.3.0: 3 points EQ.4.0: 4 points EQ.5.0: 5 points GT.5.0: valid only for trapezoidal or user defined rule	E10.0
21-30	Printout option EQ.0.0: default set to 3.0 EQ.1.0: stress resultants at element center EQ.2.0: stress resultants at plan integration points (4) EQ.3.0: stress resultants at plan integration points (4) and stresses at each integration point through the thickness at each plan point	E10.0
31-40	Quadrature rule (through thickness) LT.0.0: absolute value gives user-specified rule number EQ.0.0: Gauss (up to five points are permitted) GT.0.0: trapezoidal	E10.0

Membrane elements have no torsional or bending stiffness. Rotational degrees of freedom at membrane nodes must be constrained, either by boundary conditions or the presence of a beam or shell element.

For the same order of accuracy, Lobatto integration requires one more point than Gauss. Lobatto integration offers the advantage that its outermost points are located on the top and bottom surfaces of the shell. This is useful during postprocessing with TAURUS, which offers the *shell inner*, *shell middle*, and *shell outer* commands to select which thickness integration point's data to display. The *inner* and *outer* points are those nearest to the bottom and top surfaces of the element, respectively. The *middle* point's data is interpolated from the two center most points if an even number of points are used. Taurus *surface* stresses (components 40 and 41, for example) are calculated from resultants, and should be relatively independent of choice of integration rule.

Card 10

Columns	Quantity	Format
1-10	Shell thickness at node n_1 (see Fig. 4-5)	E10.0
11-20	Shell thickness at node n_2	E10.0
21-30	Shell thickness at node n_3	E10.0
31-40	Shell thickness at node n_4	E10.0
41-50	Location of reference surface EQ. 1.0: top surface EQ. 0.0: middle surface EQ. -1.0: bottom surface	E10.0

The thicknesses can be overridden on the element cards, i.e., the above values are used if and only if the thicknesses are zero on the element cards.

The location of the reference surface can be crucial whenever beams, shells, and solids are interconnected. Often it is desirable to have the beams and shells share a common reference surface. Whenever shells lie along sliding interfaces the reference surface should be on the contact surface.

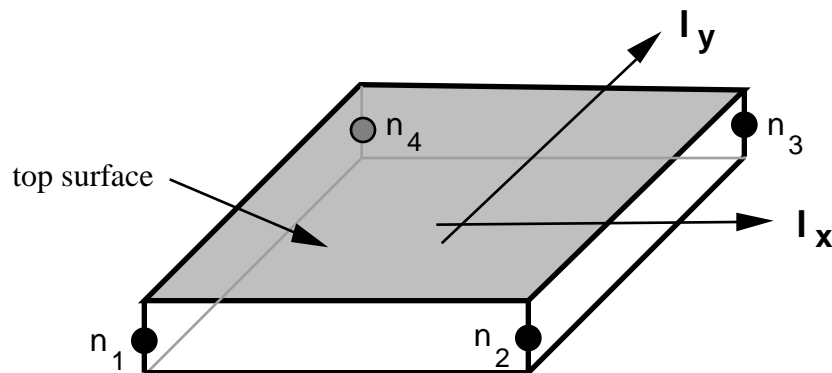


Figure 4-5. NIKE3D shell element. Node numbering determines the top surface, and orientation of local axes l_x and l_y . These axes are used for resultant force and moment calculations reported in the local coordinate system.

4.3 User-Defined Integration Rules for Beams

Define *NUSBIR* card sets in this section.

Card 1

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-5	Number of integration points, <i>NIP</i>	I5
6-10	Relative area of cross-section (the actual area divided by the area computed from cross-section dimensions given in the material deck)	E10.0

Cards 2, 3, ..., (<i>NIP</i>+1)
--

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-10	<i>s</i> -coordinate of integration point	E10.0
11-20	<i>t</i> -coordinate of integration point	E10.0
21-30	Weighting factor. Typically, the area associated with the integration point divided by the total cross sectional area is used	E10.0

4.4 User-Defined Integration Rules for Shells

Define *NUSSIR* card sets in this section.

Card 1

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-5	Number of integration points, <i>NIP</i>	I5
6-10	Integration point spacing option, <i>IESOPT</i> EQ. 0: integration points are defined below EQ. 1: integration points are equally spaced through the thickness such that the shell is divided into <i>NIP</i> layers of equal thickness.	I5

(Define the following cards only if *IESOPT*=0 above)

Cards 2, 3, ... , (<i>NIP</i>+1)

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-10	Coordinate of integration point (−1 1)	E10.0
11-20	Weighting factor. Typically, the area associated with the integration point divided by the total cross sectional area is used	E10.0

4.5 Node Point Deck

Cards 1, 2, ... , NUMNP

Columns	Quantity	Format
1-8	Node number	I8
9-13	Displacement boundary condition code EQ.0: no constraints EQ.1: constrained x displacement EQ.2: constrained y displacement EQ.3: constrained z displacement EQ.4: constrained x and y displacements EQ.5: constrained y and z displacements EQ.6: constrained z and x displacements EQ.7: constrained x, y, and z displacements	I5
14-33	x-coordinate	E20.0
34-53	y-coordinate	E20.0
54-73	z-coordinate	E20.0
74-78	Rotational boundary condition code EQ.0: no constraints EQ.1: constrained x rotation EQ.2: constrained y rotation EQ.3: constrained z rotation EQ.4: constrained x and y rotations EQ.5: constrained y and z rotations EQ.6: constrained z and x rotations EQ.7: constrained x, y, and z rotations	I5

Whenever node data is missing, nodes are automatically generated according to the sequence $[n_i, (n_i + 1), (n_i + 2), \dots, n_j]$, where n_i and n_j are the node numbers defined on two consecutive cards. Linear interpolation is used to obtain the coordinates of the generated nodes. The boundary condition code is set to zero whenever the boundary condition code of n_i differs from that of n_j . The first node and the last node must always be defined.

Constrained degrees of freedom are removed from the calculation, hence reaction forces are not computed. To obtain the reaction force at a fixed node, use a displacement boundary condition with zero value. See the displacement boundary condition deck for more information.

4.6 Hexahedron Element Deck

Cards 1, 2, ... , *NUMELH*

Columns	Quantity	Format
1-8	Element number	I8
9-13	Material number	I5
14-21	Node point n_1	I8
22-29	Node point n_2	I8
30-37	Node point n_3	I8
• •		
• •		
• •		
• •		
70-77	Node point n_8	I8

Node points n_1 - n_8 define the corner nodes of the 8-node solid elements. Degenerated elements having fewer than 8 nodes are obtained by repeating one or more nodes. Input of nodes on the element cards for the four and six noded elements shown in Fig. 2-1 takes the form:

4-node n_1 n_2 n_3 n_4 n_4 n_4 n_4 n_4

6-node n_1 n_2 n_3 n_4 n_5 n_5 n_6 n_6

Element cards are assumed to be in element number sequence. Omitted data is automatically generated with respect to the first card prior to the omitted data. Node numbers are generated using the sequence $n_j^{i+1} = n_j^i + 1$, ($j = 1, 8$) The material properties for the generated elements are taken from the first card prior to the omitted data.

4.7 Beam Element Deck

Cards 1, 2, ... , *NUMELB*

Columns	Quantity	Format
1-8	Element number	I8
9-13	Material number	I5
14-21	Node point n_1 (see Fig. 4-6)	I8
22-29	Node point n_2	I8
30-37	Node point n_3 (to determine orientation)	I8
38-45	Cross section dimension d_{s1} at node n_1 (optional) rectangular: d_{s1} = width in s direction (see Fig. 4-6) tubular: d_{s1} = outside diameter	E8.0
46-53	Cross section dimension d_{s2} at node n_2 (optional) rectangular: d_{s2} = width in s direction tubular: d_{s2} = outside diameter	E8.0
54-61	Cross section dimension d_{t1} at node n_1 (optional) rectangular: d_{t1} = width in t direction tubular: d_{t1} = inside diameter	E8.0
62-69	Cross section dimension d_{t2} at node n_2 (optional) rectangular: d_{t2} = width in t direction tubular: d_{t2} = inside diameter	E8.0

Element cards are assumed to be in element number sequence. Omitted data is automatically generated with respect to the first card prior to the omitted data as follows:

$$n_j^{i+1} = n_j^i + 1, \quad (j = 1, 2)$$

The material properties, cross-sectional properties, and orientation node, n_3 , for the generated elements are taken from the card preceding the generated data. If the thicknesses are undefined, they are taken from the material cross section definition cards.

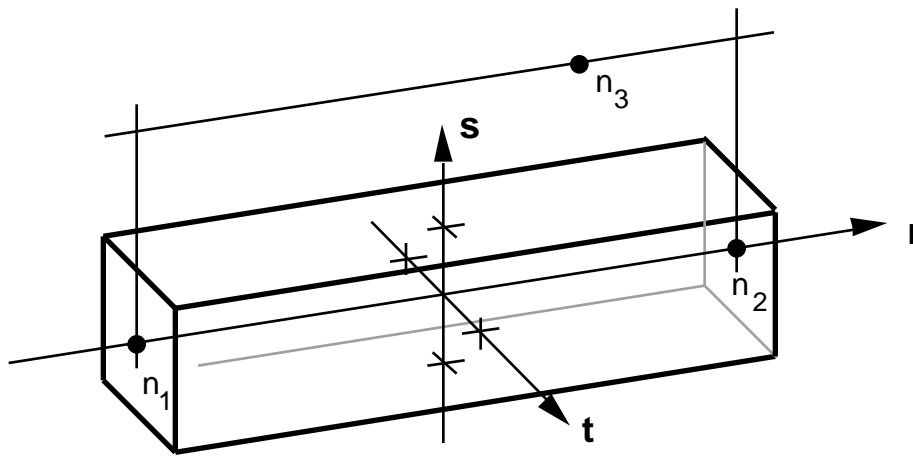


Figure 4-6. NIKE3D beam element node n_3 determines the initial orientation.

4.8 Shell Element Deck

Define two cards for each shell element:

Cards 1, 2, ... , 2*NUMELS

Columns	Quantity	Format
First Card		
1-8	Element number	I8
9-13	Material number	I5
14-21	Node point n_1 (see Fig. 6)	I8
22-29	Node point n_2	I8
30-39	Node point n_3	I8
40-45	Node point n_4	I8
Second Card		
1-10	Shell thickness at node 1 (optional)	E10.0
11-20	Shell thickness at node 2 (optional)	E10.0
21-30	Shell thickness at node 3 (optional)	E10.0
31-40	Shell thickness at node 4 (optional)	E10.0
41-50	Orthotropic material angle, , degrees (optional)	E10.0

Currently, three shell element formulations are available: the Hughes-Liu (default), YASE, and Belytschko-Tsay. The shell element formulation is specified on Control Card 8. Element cards are assumed to be in element number sequence. Omitted data is automatically generated with respect to the first card prior to the omitted data as follows:

$$n_j^{i+1} = n_j^i + 1, \quad (j = 1, 4)$$

where n_j^{i+1} is the node number for node j of element $i+1$. The material and cross-sectional properties for the generated elements are taken from the card preceding the generated data. If the thicknesses are zero, they are taken from the material cards.

4.9 Rigid Node and Facet Deck

Define the number of cards, *NUMRNF* specified on Control Card 2.

Cards 1, 2, ..., <i>NUMRNF</i>

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-5	Rigid body number	I5
6-13	Node point n_1	I8
14-21	Node point n_2	I8
22-29	Node point n_3	I8
30-37	Node point n_4	I8

Rigid nodes and facets move as a part of the indicated rigid body. The input is a very general list of node points, four (or less) per card. The nodes need not be unique, and need not form a facet of an element, although this method of input is often convenient. This feature gives a convenient way to attach a portion of a deformable mesh to a rigid body without merging node points or using a tied interface.

4.10 Discrete Element Deck

Skip this section if the discrete element flag, *INPDE*, on control card 2 is zero. Discrete elements offer the ability to apply complex boundary conditions at single node points. The discrete element deck is organized similar to the entire NIKE3D input file: A discrete element control card is defined which contains information about the data to follow. Discrete element material definitions appear next, if necessary, followed by definitions of the discrete springs, dampers, and masses as required.

Discrete Element Control Card

Columns	Quantity	Format
1-5	Number of discrete material definitions, <i>NDEMAT</i>	I5
6-10	Number of discrete springs and dampers, <i>NUMDEL</i>	I5
11-15	Number of discrete masses, <i>NUMMAS</i>	I5

Discrete Element Material Deck

Define *NDEMAT* discrete element materials below. Each material is defined by a set of two cards. Card 1 is common to all material types. The format of Card 2 varies with each material.

Card 1

Columns	Quantity	Format
1-5	Discrete material number (<i>NDEMAT</i>)	I5
6-10	Discrete material type: I5 EQ.1: linear elastic EQ.2: linear viscous EQ.3: isotropic elastoplastic EQ.4: nonlinear elastic EQ.5: nonlinear viscous EQ.6: nonlinear general	

Material Type 1 for Discrete Elements (Linear Elastic)

<u>Columns</u>		<u>Quantity</u>	<u>Format</u>
1-10	Card 2	Elastic stiffness (force/displacement)	E10.0

Material Type 2 for Discrete Elements (Linear Viscous)

<u>Columns</u>		<u>Quantity</u>	<u>Format</u>
1-10	Card 2	Damping constant (force/displacement rate)	E10.0

Material Type 3 for Discrete Elements (Isotropic Elastoplastic)

<u>Columns</u>		<u>Quantity</u>	<u>Format</u>
1-10	Card 2	Elastic stiffness (force/displacement)	E10.0
11-20		Tangent stiffness (force/displacement)	E10.0
21-30		Yield (force)	E10.0
31-40		Hardening parameter $0 \quad 1$.eq.0: kinematic hardening .eq. 1: isotropic hardening	E10.0

Material Type 4 for Discrete Elements (Nonlinear Elastic)

<u>Columns</u>		<u>Quantity</u>	<u>Format</u>
1-10	Card 2	Load curve number describing force versus displacement relationship	E10.0

Material Type 5 for Discrete Elements (Nonlinear Viscous)

Columns	Quantity		Format
1-10	Card 2	Load curve number describing force versus rate-of-displacement relationship	E10.0

Material Type 6 for Discrete Elements (General Nonlinear)

Columns	Quantity		Format
1-10	Card 2	Load curve number describing force versus displacement relationship for loading	E10.0
11-20		Load curve number describing force versus displacement relationship for unloading	E10.0
21-30		Hardening parameter ($0 < \quad < 1$)	E10.0
31-40		Initial yield force in tension (> 0)	E10.0
41-50		Initial yield force in compression (< 0)	E10.0

Load curves describing force-displacement behavior are used by interpreting the time value as displacement, and the function value as force. The points must be in order starting with the most negative (compressive) displacement and ending with the most positive (tensile). The curves need not be symmetric.

The displacement origin of the "unloading" curve is arbitrary, since it will be shifted as necessary as the element extends and contracts. On reverse yielding the "loading" curve will also be shifted along the displacement axis. The initial tensile and compressive yield forces (FYT and FYC) define a range within which the element remains elastic (i.e. the "loading" curve is used for both loading and unloading). If at any time the force in the element exceeds this range, the element is deemed to have yielded, and at all subsequent times the "unloading" curve is used for unloading.

The hardening parameter is interpreted like that of material type 3. For $=0$, kinematic-like hardening behavior is achieved, and the elastic range of the element remains fixed. For $=1$, isotropic-like hardening behavior is achieved, and the elastic range evolves with the loading.

Discrete Spring and Damper Definitions

Define *NUMDEL* discrete springs and dampers below.

Cards 1, 2, ... , <i>NUMDEL</i>
--

Columns	Quantity	Format
1-5	Element number	I5
6-13	Node, n_1	I8
14-21	Node, n_2	I8
22-26	Material number	I5
27-36	Scale factor on force (default = 1.0)	E10.0

Discrete Mass Definitions

Define *NUMMAS* lumped masses below.

Cards 1, 2, ... , <i>NUMMAS</i>
--

Columns	Quantity	Format
1-8	Node number	I8
9-18	Mass	E10.0

4.11 Constrained Node Pairs Deck

Cards 1, 2, ... , NRCC

Columns	Quantity	Format
1-8	First node of pair	I8
9-16	Second node of pair	I8
17-21	Constrained translational degrees of freedom EQ.0: no translational constraints EQ.1: constraint in x EQ.2: constraint in y EQ.3: constraint in z EQ.4: constraint in x and y EQ.5: constraint in y and z EQ.6: constraint in z and x EQ.7: constraint in x, y, and z	I5
22-26	Constrained rotational degrees of freedom EQ.0: no rotational constraints EQ.1: constraint in x EQ.2: constraint in y EQ.3: constraint in z EQ.4: constraint in x and y EQ.5: constraint in y and z EQ.6: constraint in z and x EQ.7: constraint in x, y, and z	I5

Selected degrees of freedom of the node pairs listed above are constrained to move together.

Global degrees of freedom are used. The first node of the pair is the master node; therefore, when defining multiple nodes to have the same degrees of freedom, the same node should be used as the master node.

4.12 1D Slideline Deck

The 1D Slideline Deck consists of *NUMID* control cards, one for each interface. These cards are followed by a set of slave and master segment definitions for each interface. The number of slave and master segments defined for each interface is taken from that interface's control card.

1D Slideline Control Cards
1, 2, ... , *NUMID*

Columns	Quantity	Format
1-8	Number of slave (beam) segments (<i>NSS_i</i>)	I8
9-16	Number of master (solid) segments (<i>NMS_i</i>)	I8
17-20	Algorithm type flag EQ.1: two pass (DEFAULT) EQ.2: one pass, slave onto master EQ.3: one pass, master onto slave	I4
21-30	Penalty stiffness multiplier (lateral directions) EQ.0.0: default set to "1.0"	E10.0
31-40	Penalty stiffness multiplier (axial direction) EQ.0.0: default set to "1.0"	E10.0
41-50	Axial bond strength, <i>S</i>	E10.0
51-55	Initialization node relocation flag EQ.1: adjust for uniform contact (DEFAULT) EQ.2: reposition slave nodes only EQ.3: reposition master nodes only EQ.4: initial geometry unchanged	I5
56-65	Axial constraint birth time (<i>TBIRTH</i>) EQ.0: axial constraints always active NE.0 frictionless axial sliding until $t = TBIRTH$, constrained thereafter	E10.0
66-70	Augmentation flag <i>IAUG</i> <i>IAUG</i> < 0: augment exactly ($-IAUG$) times per step <i>IAUG</i> =0: no augmentations (penalty method) <i>IAUG</i> =1: augment until convergence tolerance satisfied	I5
71-80	Convergence tolerance for augmentations, <i>ALTOL</i> GT.0.: converged when force norm < <i>ALTOL</i> EQ.0.: DEFAULT = 0.01	E10.0

Define the set of slave and master segment cards described below. Repeat this set for each interface i ($1 \leq i \leq NUMID$). The number of cards in each set ($NSS_i + NMS_i$) is taken from the corresponding 1D slideline control card. Missing segments are generated automatically using the generation increment from the first segment in the sequence.

Slave Segment Cards $1, 2, \dots, NSS_i$
--

Columns	Quantity	Format
1-8	Slave facet number	I8
9-16	Generation increment	I8
17-24	Node point n_1	I8
25-32	Node point n_2	I8

Master Segment Cards $(NSS_i+1), (NSS_i+2), \dots, (NSS_i+NMS_i)$

Columns	Quantity	Format
1-8	Master facet number	I8
9-16	Generation increment	I8
17-24	Node point n_1	I8
25-32	Node point n_2	I8

The 1D slideline acts to constrain beam or truss elements to slide along the edges of brick elements, and is fully documented in Maker and Laursen [1994]. An obvious application is reinforced concrete, where the beam elements represent steel rebar imbedded in a matrix of concrete (solid elements). Two constraints are imposed by the slideline: one in the lateral direction (rebar must follow original "hole" in concrete), and one in the axial direction (rebar may pull out if bond strength is exceeded). Overlapping interface definitions are permitted, as in the case where the rebar elements extend beyond the surface of the concrete.

The beam elements must be chosen as the slave side of the interface. A symmetric *two-pass algorithm* is used by default to constrain slave nodes to follow master segments, and also master nodes to follow slave segments. This method can create "locking" in curved geometries, so alternative *single pass algorithms* are available to constrain only one set of nodes to follow its opposing set of segments. Generally the more finely zoned side should be "put onto" the more coarsely zoned side.

Constraints are imposed by a penalty method, where NIKE3D chooses an initial penalty stiffness based on geometric and material properties of the beam and solid elements. The *penalty stiffness multipliers* allow scaling of NIKE3D's stiffness values for the lateral and axial constraints independently.

The *axial bond strength* may be used to delay axial sliding until bond stress exceeds a threshold value. The axial constraint force at each beam node is divided by the wetted area it represents to obtain an axial "stress", which is compared to the bond strength. Representative wetted area is taken as the circumference times length of the beam element. In this manner, large diameter "rebar" has more holding power than small rebar with equal bond strength. Frictionless sliding is achieved by setting the bond strength to zero.

Axial constraints become active after problem time reaches the *axial constraint birth time TBIRTH*. Prior to this time frictionless axial sliding is permitted. This feature is useful to apply preload to an imbedded rebar or bolt shank.

The node relocation feature adjusts the initial geometry of the mesh to ensure a stress-free initial state in problems where faceted curves yield initial gaps across slidelines. A summary is printed to the output file. The *initialization node relocation flag* may be used to suppress node relocation.

The *augmentation* option is an alternative to the penalty method for enforcing the lateral interface constraints. The method is iterative, and is applied after convergence is obtained with the standard penalty method. A fixed number of augmentations may be requested, or the force norm may be used to control convergence. Convergence of the augmentation procedure assures the accuracy of the lateral constraint to within the user specified tolerance, independent of penalty stiffness value. Progress of the augmentation loops may be monitored by activating sense switch **SW3**. See the overview section for further details.

4.13 Sliding Surface Deck

The Sliding Surface Deck consists of at least *NUMSI* control cards, one for each interface. An additional control card may be included for an interface by activating the auxiliary interface control card flag. These cards are followed by a set of slave and master facet definitions for each interface. The number of slave and master facets defined for each interface is taken from that interface's control card.

Interface Control Cards
1, 2, ... , *NUMSI*

Columns	Quantity	Format
1-8	Number of slave facets (<i>NSSI</i>)	I8
9-16	Number of master facets (<i>NMSI</i>)	I8
17-20	Interface type Type 1 - frictionless sliding only (no gaps) Type 2 - tied Type 3 - frictional sliding with gaps Type 4 - Single surface	I4
21-30	Scale factor for sliding surface penalties EQ.0.0: default set to "1.0"	E10.0
31-40	Static friction coefficient μ_s	E10.0
41-50	Kinetic friction coefficient μ_k	E10.0
51-60	Friction decay exponent m	E10.0
61-70	Small penetration search distance, <i>PEND</i> EQ.0: contact constraints apply to all penetrated nodes GT.0: nodes penetrated farther than <i>PEND</i> are ignored	E10.0
71-75	Bandwidth minimization radius, <i>BWRAD</i> EQ.0: all potential connectivity is considered GT.0: initial gaps greater than <i>BWRAD</i> element lengths are unconnected during bandwidth minimization	I5
76-80	Auxiliary Interface Control Card flag <i>AICC</i> EQ.1: read an Auxiliary Interface Control Card immediately following this card	I5

Auxiliary Interface Control Card
(include only if $AICC=1$ on previous control card)

Columns	Quantity	Format
1-5	Augmentation flag $IAUG$ $IAUG < 0$: augment exactly ($-IAUG$) times per step $IAUG=0$: no augmentations (penalty method) $IAUG=1$: augment until convergence tolerance satisfied	I5
6-15	Normal direction convergence tolerance for augmentations, $ALTOLN$ GT.0.: converged when force norm $< ALTOLN$ EQ.0.: DEFAULT = 0.05 LT.0.: converged when gap norm $< ALTOLN $	E10.0
16-25	Tangential direction convergence tolerance for augmentations, $ALTOLT$ GT.0.: converged when force norm $< ALTOLT$ EQ.0.: DEFAULT = 0.05 LT.0.: converged when gap norm $< ALTOLT $	E10.0
26-35	Tangent stiffness multiplier $TKMULT$ EQ.0: DEFAULT = 0.001	E10.0
36-45	Interface death time (DEFAULT=1.0e+30)	E10.0
46-55	Interface burial time (DEFAULT=1.0e+30)	E10.0

Define the set of slave and master facet cards described below. Repeat this set for each interface i ($1 \leq i \leq NUMSI$). The number of cards in each set ($NSS_i + NMS_i$) is taken from the corresponding interface control card.

Slave Facet Cards 1, 2, ... , NSS_i

Columns	Quantity	Format
1-8	Slave facet number	I8
9-16	Node point n_1	I8
17-24	Node point n_2	I8
25-32	Node point n_3	I8
33-40	Node point n_4	I8

Master Facet Cards $(NSS_i+1), (NSS_i+2), \dots, (NSS_i+NMS_i)$

Columns	Quantity	Format
1-8	Master facet number	I8
9-16	Node point n_1	I8
17-24	Node point n_2	I8
25-32	Node point n_3	I8
33-40	Node point n_4	I8

Slave and master facet cards are assumed to be in sequence, though the particular number assigned to a master facet is arbitrary. Omitted data is automatically generated with respect to the first card prior to the omitted data as

$$n_j^{i+1} = n_j^i + k$$

where n_j^{i+1} is node j for facet $i+1$. The generation parameter k is taken from the first card.

Sliding Surface Notes:

Node numbering of facets in solid elements can be either clockwise or counterclockwise. For facets contained in shell elements, however, the numbering must be counterclockwise such that when applying the right-hand-rule the thumb points towards the surface the facet will potentially contact (Fig. 4-7). Triangular facets are defined by repeating a node. Every slave and master facet in the contacting surfaces must be defined. No ordering is assumed or expected in the definition of the surfaces.

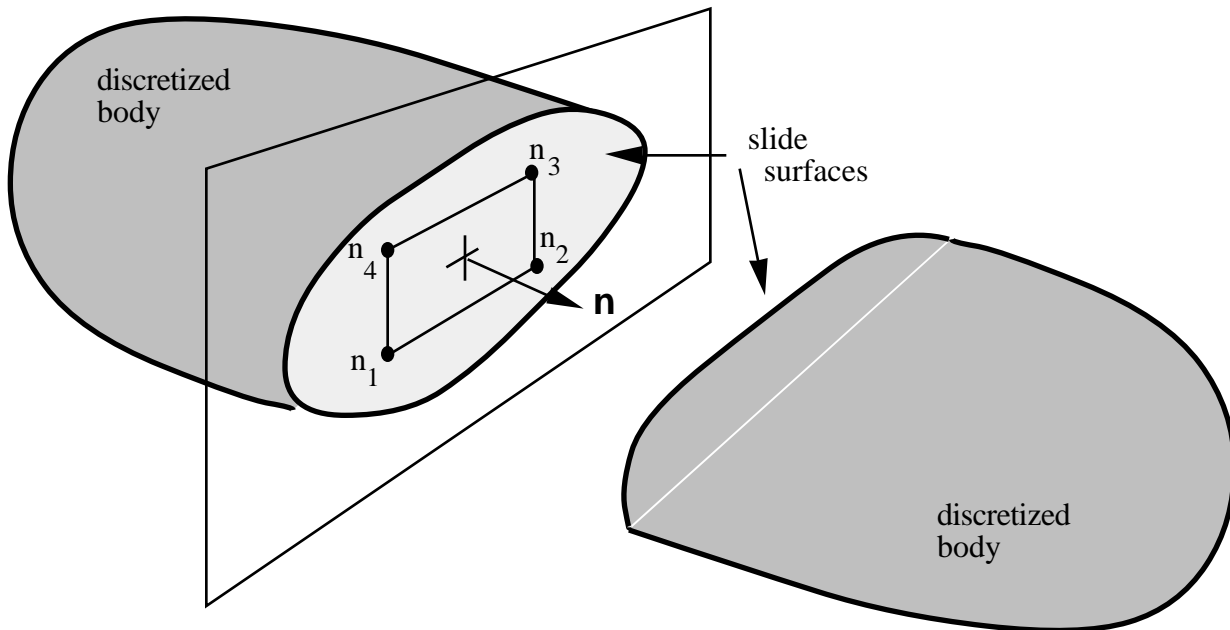


Figure 4-7. Node numbering convention for slide surface facets. Vector \mathbf{n} is the normal vector pointing toward the opposing body. Its sign is determined using the right-hand-rule from facet edge $\bar{n}_1\bar{n}_2$ to edge $\bar{n}_1\bar{n}_4$. Note that this convention *must* be followed for bodies discretized using shell elements.

The *friction model* is similar to the model for an elastic-plastic material. Corresponding to Young's modulus and the strain are the surface stiffness and the relative tangential displacement between a node and the surface facet it contacts. The product of the displacement dependent coefficient of friction and the normal force corresponds to the yield stress. An exponential function is used to smooth the transition between the static and kinetic coefficients of friction:

$$\mu = \mu_k + (\mu_s - \mu_k) e^{-md_{\text{relative}}}$$

Buckling analysis requires the *single surface contact* algorithm to model a surface collapsing onto itself. Simply using type 3 with the slave facets the same as the master facets will not work. The single surface algorithm is significantly more expensive to use than the master-slave algorithm. Its use should be restricted to those situations where it is absolutely necessary.

The *bandwidth minimization radius* determines the extent to which the bandwidth minimizer anticipates potential contact across the interface. By default, connectivity across a sliding surface is anticipated for every node, regardless of the presence of gaps in the initial geometry. This can create overly large bandwidth minimization problems. Entering a radius of n causes potential contacts with initial gaps larger than n element lengths to be ignored. To include only those portions of the interface which are actually in contact, choose a negative radius. This parameter effects only the bandwidth minimization phase of the analysis, not the solution phase.

The *augmentation* option is an alternative to the penalty method for enforcing contact constraints. The method is iterative, and is applied after convergence is obtained with the standard penalty method. A fixed number of augmentations may be requested, or the force or gap norms may be used to control convergence. Convergence of the augmentation procedure assures accuracy of the constraints to within the user specified tolerance, independent of penalty stiffness value. Progress of the augmentation loops may be monitored by activating sense switch **SW3**. See the overview section for further details.

The *tangent stiffness multiplier* adjusts the tangential stiffness of the contact interface during the slip phase of the Coulomb stick-slip process. It is used only with slide surface type 3. Ideally, once the stick condition has been broken, sliding occurs freely hence with zero tangential stiffness. Practically, zero stiffness is acceptable only when rigid body motion is constrained. For force-driven, quasi-static problems, unrestrained rigid body motion is often possible once sliding begins. In these problems, some tangential stiffness is required to eliminate a singularity in the resulting linearized equilibrium equations. The default value provides such stiffness. Improved convergence may be obtained for dynamic and displacement driven quasi-static problems by reducing the tangent stiffness multiplier to near zero ($TKMULT = 1.0e-10$).

Vanishing contact interfaces are activated by specifying death and burial times. At time of death, the interface begins to vanish from the problem, and is completely gone at burial time. This feature allows interfaces to be slowly removed from a model. It is useful, for example, in metal forming springback calculations to remove tooling from the model.

4.14 Nodal Reorder Data

Include this data if the Bandwidth Minimization Flag on Control Card #6 indicates that the nodal reorder vector is to be read from the input deck. This feature may not be used with models that contain rigid bodies.

Nodal Reorder Data

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-10	Sequence number for node 1	I10
11-20	Sequence number for node 2	I10
.		
.		
71-80	Sequence number for node 8	I10

This option allows the results of an (external) bandwidth minimizer to be used in place of NIKE3D's GPS routines. NIKE3D will assign internal equation numbers to nodal degrees of freedom in the sequence defined above. This renumbering is intended to optimize performance of the direct linear equation solvers. Since equation numbers are used, the user node numbers are not changed, and the operation is transparent to the user.

4.15 Stonewall and Symmetry Plane Deck

Define *NRWSP* stonewalls/symmetry planes as specified on Control Card 2. Repeat the following set of cards for each stonewall or symmetry plane. The auxiliary control card need not be included if the default values for its parameters are used.

Stonewall / Symmetry Plane Control Card
--

Columns	Quantity	Format
1-5	Number of slave nodes, <i>NSN</i> LT.0: symmetry plane, then $NSN = NSN $	I5
6-15	x-coordinate of tail of any outward normal vector \mathbf{v} (See Fig. 4-8) (node n_1 for <i>following</i> symmetry plane)	E10.0
16-25	y-coordinate of tail (node n_2 for <i>following</i> symmetry plane)	E10.0
26-35	z-coordinate of tail (node n_3 for <i>following</i> symmetry plane)	E10.0
36-45	x-coordinate of head (node n_1 for <i>following</i> symmetry plane)	E10.0
46-55	y-coordinate of head (node n_2 for <i>following</i> symmetry plane)	E10.0
56-65	z-coordinate of head (node n_3 for <i>following</i> symmetry plane)	E10.0
66-75	Penalty stiffness scale factor EQ. 0.0: default = 1.0	E10.0
76-80	Auxiliary Control Card flag <i>AICC</i> EQ.1: read an Auxiliary Control Card immediately following this card	I5

Stonewall / Symmetry Plane

Auxiliary Control Card

(include only if $AICC=1$ on previous control card)

Columns	Quantity	Format
1-5	Load curve giving displacement history of moving stonewall EQ. 0: stonewall is fixed in space	I5
6-10	Augmentation flag $IAUG$ $IAUG < 0$: augment exactly ($-IAUG$) times per step $IAUG=0$: no augmentations (penalty method) $IAUG=1$: augment until convergence tolerance satisfied	I5
11-20	Convergence tolerance for augmentations, $ALTOL$ GT.0.: converged when force norm $< ALTOL$ EQ.0.: DEFAULT = 0.01	E10.0

Stonewall / Symm. Plane Nodes

Cards 1, 2, ... , NSN

Columns	Quantity	Format
1-8	Slave node number	I8
9-16	Node point number	I8

Omitted slave nodes are automatically generated by incrementing the node point numbers by

$$\frac{n_j - n_i}{sn_j - sn_i}$$

here sn_i and sn_j are slave numbers on two successive cards and n_i and n_j are their corresponding node numbers.

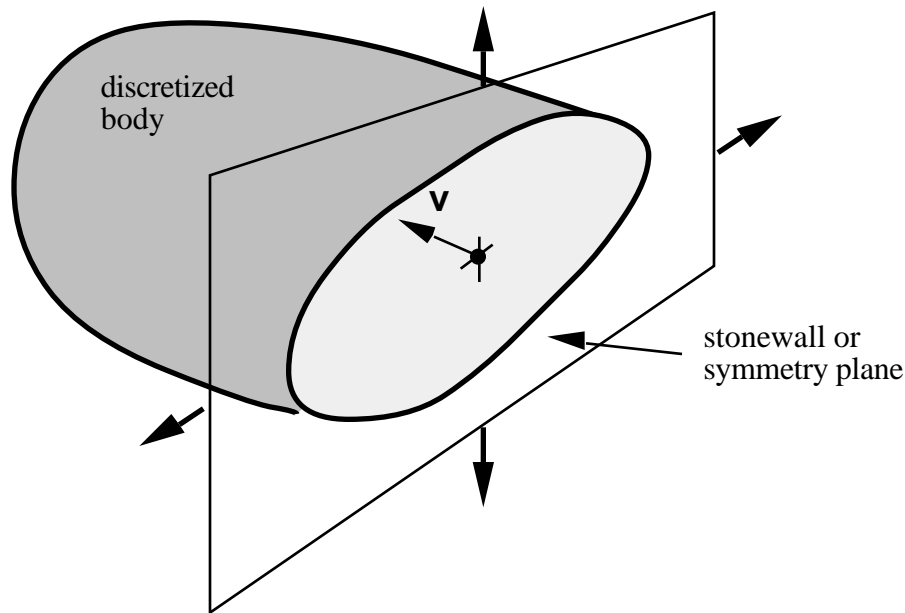


Figure 4-8. This view shows an arbitrary body with a boundary on a stonewall or symmetry plane. The plane extends to infinity in each direction. The view also depicts clearly the sign convention used in defining the orientation vector \mathbf{v} , which originates at any point on the plane, is normal to the plane, and points toward the body. For moving stonewalls, the displacement history of the stonewall in the direction of \mathbf{v} is specified using a load curve.

Stonewall / Symmetry Plane Notes:

A *stonewall* is a planar surface extending to infinity. It is defined by a normal vector of arbitrary magnitude drawn outward from the plane (see Fig. 4-8). Stonewalls act as rigid barriers. Nodes that are designated as slave nodes cannot penetrate but may separate, other nodes can penetrate. The initial location of slave nodes may be on the stonewall, or above it (in the outward direction).

Symmetry planes provide the capability for modeling off-axis displacement constraints, for example, a 30° slice of an axisymmetric object. The slave nodes must lie in the symmetry plane in the undeformed configuration, and may only move tangentially along the plane (no separation).

Following symmetry planes use three node points to define the plane. The plane's position is continually updated as these three nodes move. Boundary conditions may be applied to these three nodes. Nodes designated as slave nodes must remain in the plane, but may move frictionlessly within it. The three defining nodes may not be listed as slave nodes. This provides the capability, for example, to enforce the constraint that a cross section remain planar as it moves in space. To invoke the following symmetry plane, enter the three node numbers in place of the coordinates of *both the head and tail* of vector \mathbf{v} .

If a non-zero load curve is specified, the displacement time history of the stonewall is given by the load curve. Displacements are prescribed in the direction of the normal vector \mathbf{v} . Users should exercise caution when retracting stonewalls away from objects, since temporary loss of contact with the stonewall may introduce rigid body modes to the model.

Stonewalls and symmetry planes are implemented using a penalty method. Under some circumstances, interpenetrations or stress discontinuities across symmetry planes may be observed. In these cases the penalty stiffness multiplier may be increased by one or more orders of magnitude, at the potential cost of increasing convergence iterations in the nonlinear solver.

The *augmentation* option is an alternative to the penalty method for enforcing the impenetrability condition for stonewalls only. The method is iterative, and is applied after convergence is obtained with the standard penalty method. A fixed number of augmentations may be requested, or the force norm may be used to control convergence. Convergence of the augmentation procedure assures the accuracy of the impenetrability constraint to within the user specified tolerance, independent of penalty stiffness value. Progress of the augmentation loops may be monitored by activating sense switch **SW3**. See the overview section for further details.

The *total reaction forces* computed for each stonewall are included in the printed and plotted output. These forces are expressed as vector components in the global coordinate system. They may be viewed in phase two of TAURUS with the *stime* command, using component numbers generated by NIKE3D. These numbers uniquely identify global force components for each interface in the model, hence they vary from model to model. NIKE3D includes a tabulated list of component numbers in the screen output and in the printed output (default filename *n3dhsp*).

4.16 Node Print Blocks

Skip this section if the number of node print blocks, *NNPB*, on Control Card 5 is zero. Otherwise, include one card with the following information. Up to five print blocks may be defined. Only these nodes will be printed in the high speed printer file.

Card 1

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-8	First node of first print block	I8
9-16	Last node of first print block	I8
17-24	First node of second print block	I8
25-32	Last node of second print block	I8
.		
.		
.		

4.17 Solid Element Print Blocks

Skip this section if the number of brick element print blocks, *NHPB*, on Control Card 5 is zero. Otherwise, include one card with the following information. Up to five print blocks may be defined. Only these solid elements will be printed in the high speed printer file.

Card 1

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-8	First brick element of first print block	I8
9-16	Last brick element of first print block	I8
17-24	First brick element of second print block	I8
25-32	Last brick element of second print block	I8
.		
.		
.		

4.18 Beam Element Print Blocks

Skip this section if the number of beam element print blocks, *NBPB*, on Control Card 5 is zero. Otherwise, include one card with the following information. Up to five print blocks may be defined. Only these beam elements will be printed in the high speed printer file.

Card 1

Columns	Quantity	Format
1-8	First beam element of first print block	I8
9-16	Last beam element of first print block	I8
17-24	First beam element of second print block	I8
25-32	Last beam element of second print block	I8
.		
.		
.		

4.19 Shell Element Print Blocks

Skip this section if the number of shell element print blocks, *NSPB*, on Control Card 5 is zero. Otherwise, include one card with the following information. Up to five print blocks may be defined. Only these shell elements will be printed in the high speed printer file.

Card 1

Columns	Quantity	Format
1-8	First shell element of first print block	I8
9-16	Last shell element of first print block	I8
17-24	First shell element of second print block	I8
25-32	Last shell element of second print block	I8
.		
.		
.		

4.20 Load Curve Deck

"Load" curves are simple tabulations of amplitude vs. time data. They are used to specify the magnitude and/or time variation of many types of data, including boundary conditions and material properties. Any boundary condition or property may reference any load curve, and a load curve may be referenced more than once. NIKE3D checks for initial nonzero applied load by evaluating load curve data at time zero, and performs a complete nonlinear solution at time zero (step #0) if necessary. Applying load at time zero is not recommended, since no computational advantage is gained, and the automatic time step controller cannot intervene if equilibrium convergence is not obtained.

In this deck, define the number of load curves, *NUMLC*, specified on Control Card 4. Repeat the following cards for each curve:

Card 1

Columns	Quantity	Format
1- 5	Load curve number	I5
6-10	Number of points in load curve, <i>NPTS</i>	I5

Card 2, ... , (*NPTS*+1)

Columns	Quantity	Format
1-10	Time	E10.0
11-20	Load curve value	E10.0

If the current time during an analysis exceeds the largest time for which a load curve is defined, a warning message is printed and NIKE3D *extrapolates a load value based on the last segment of the load curve*.

4.21 Concentrated Nodal Loads Deck

Define the number of concentrated nodal loads, *NUMCNL*, specified on Control Card 4.

Cards 1, 2, ... , <i>NUMCNL</i>
--

Columns	Quantity	Format
1-8	Node point number on which this load acts	I8
9-13	Direction in which load acts EQ.1: x-direction force EQ.2: y-direction force EQ.3: z-direction force EQ.4: moment about x-direction EQ.5: moment about y-direction EQ.6: moment about z-direction	I5
14-18	Load curve number	I5
19-28	Scale factor EQ.0.0: default set to "1.0"	E10.0

Concentrated nodal loads are defined in global coordinates. In contrast to "follower forces", concentrated nodal loads act in a fixed direction as an analysis proceeds. Hence, loads that were defined normal to a surface in the undeformed configuration may not remain normal if the surface undergoes large deformation. Pressure loads may be used to maintain normality during large deformations.

4.22 Temperature Profile Deck

Skip this section if the temperature profile input flag, *ITPRO*, on control card 7 is zero.

Cards 1, 2, ..., NUMNP

Columns	Quantity	Format
1-8	Node number I5	I8
9-18	Temperature component scaled by load curve, T_{mod}	E10.0
19-28	Temperature component constant w.r.t. time, T_{base}	E10.0
29-33	Generation increment, k	I5

If the temperature profile deck is used, a temperature will be generated for each node in the model. The first and last nodes must be defined in the input. Data for omitted nodes will be generated automatically using the generation increment, k , which defaults to a value of one.

The temperature profile data is used in conjunction with the thermal analysis option, *ITEO*, specified on control card 7. When *ITEO* is greater than zero, the thermal profile is evaluated at time t using:

$$\mathbf{T} = \mathbf{T}_{\text{mod}} * f(t) + \mathbf{T}_{\text{base}}$$

where \mathbf{T}_{mod} and \mathbf{T}_{base} are input above, and $f(t)$ is the current value of load curve *ITEO*.

When *ITEO* is less than 0, the value for $f(t)$ is given by load curve *ITPRO*. The above formula is used to evaluate the temperature profile *at time zero*. Temperatures for future times are taken from the TOPAZ input files according to *ITEO*.

4.23 Pressure Boundary Condition Deck

Define the number of cards, *NUMPR*, specified on Control Card 4.

Cards 1, 2, ..., <i>NUMPR</i>

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-5	Load curve number	I5
6-13	Node point n_1 (see Fig. 4-9)	I8
14-21	Node point n_2	I8
22-29	Node point n_3	I8
30-37	Node point n_4	I8
38-47	Multiplier of load curve at node n_1 EQ.0.0: default value is "1.0"	E10.0
48-57	Multiplier of load curve at node n_2 EQ.0.0: default value is "1.0"	E10.0
58-67	Multiplier of load curve at node n_3 EQ.0.0: default value is "1.0"	E10.0
68-77	Multiplier of load curve at node n_4 EQ.0.0: default value is "1.0"	E10.0

Pressure cards are assumed to be in sequence. Omitted data is automatically generated with respect to the first card prior to the omitted data as

$$n_j^{i+1} = n_j^i + 1$$

where n_j^i is node j of pressure card i . The load curve numbers, start times, and load curve multipliers are all taken from the first card. The load curve multipliers may be used to increase or decrease the pressure. Triangular facets are defined by repeating node n_3 .

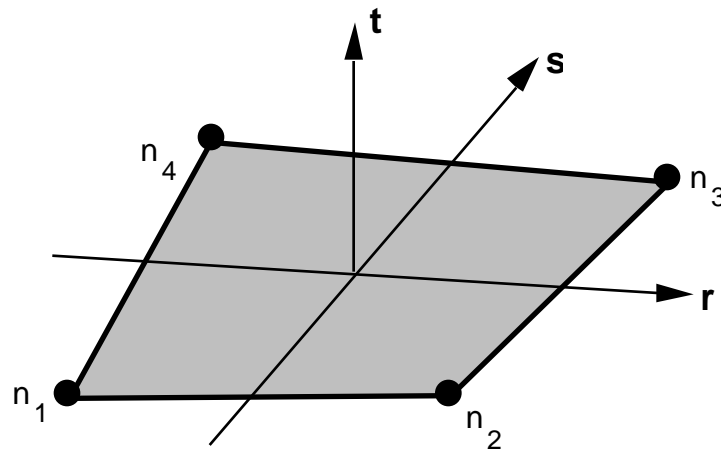


Figure 4-9. Node numbering for pressure cards. Positive pressure acts in negative t -direction.

4.24 Displacement Boundary Condition Deck

Define the number of cards, *NUMDIS*, specified on Control Card 4.

Cards 1, 2, ... , <i>NUMDIS</i>
--

Columns	Quantity	Format
1-8	Node point number to which this displacement is applied	I8
9-13	Global direction in which the node is displaced EQ.1: translation in x-direction EQ.2: translation in y-direction EQ.3: translation in z-direction EQ.4: rotation about x-direction EQ.5: rotation about y-direction EQ.6: rotation about z-direction	I5
14-18	Load curve number	I5
19-28	Scale factor on load curve EQ.0.0: default set to "1.0"	E10.0
29-33	Static Initialization Flag EQ.0: always active EQ.1: only enforced during static initialization	I5

All rotational motion is specified in units of radians. Prescribed rotations must be used at nodes which have active rotational degrees of freedom, namely those connected to beam and/or shell elements.

The static initialization flag provides a method to constrain rigid body motion during static initialization of a dynamic analysis. It is used in conjunction with analysis type option 2 on Control Card #7, *dynamic analysis with stresses initialized statically at time = 0*.

Reaction forces are now computed for nodal degrees of freedom with prescribed displacement boundary conditions. These forces are included in the printed output file, default name n3dhsp. To obtain the reaction force at a fixed node, use a displacement boundary condition whose corresponding load curve has zero values.

4.25 Base Acceleration Body Force Loads Deck

Body force loads may be applied in static or dynamic analyses. In static analyses, the model must be constrained in the direction of the acceleration at some "base" location to avoid rigid body translation. Nodal displacements, velocities, and accelerations are then computed relative to the base motion. Either relative or absolute accelerations may be selected for output using the Acceleration Data Dump Flag on Control Card 5. The sign convention for body force loads is understood by considering that the coordinate system, or *base*, is accelerated in the direction specified. Thus, a structure with its base fixed in the X-Y plane and extending upward in the +Z-direction will be crushed by a +Z acceleration, and stretched by a -Z acceleration. Material mass density must be specified for all analyses using body forces.

The Body Force due to Base Acceleration flags are specified on Control Card 4. Skip cards 1, 2, and/or 3 below if the corresponding flag (BFBAX, BFBAY, BAFBZ) is zero.

Card 1

Columns	Quantity	Format
1-5	Load curve number	I5
6-15	Scale factor on X-acceleration EQ.0.0: default set to "1.0"	E10.0

Card 2

Columns	Quantity	Format
1-5	Load curve number	I5
6-15	Scale factor on Y-acceleration EQ.0.0: default set to "1.0"	E10.0

Card 3

Columns	Quantity	Format
1-5	Load curve number	I5
6-15	Scale factor on Z-acceleration EQ.0.0: default set to "1.0"	E10.0

4.26 Angular Velocity Body Force Loads Deck

Body forces due to angular velocity act radially outward from the axis of rotation, and are always calculated with respect to the deformed configuration. Angular velocity is assumed to have the units of radians per unit time. Torsional effects, generated for example by increasing angular velocity at a constant rate, are not included. Material mass density must be specified for all analyses using body forces

The Body Force due to Angular Velocity flags are specified on Control Card 4. Skip cards 1, 2, and/or 3 below if the corresponding flag (BFAVX, BFAVY, BFAVZ) is zero.

Card 1

Columns	Quantity	Format
1-	Load curve number	I5
6-15	Scale factor on angular velocity about X-axis EQ.0.0: default set to "1.0"	E10.0

Card 2

Columns	Quantity	Format
1-5	Load curve number	I5
6-15	Scale factor on angular velocity about Y-axis EQ.0.0: default set to "1.0"	E10.0

Card 3

Columns	Quantity	Format
1-5	Load curve number	I5
6-15	Scale factor on angular velocity about Z-axis EQ.0.0: default set to "1.0"	E10.0

4.27 Initial Velocity Deck

Skip this section if the initial velocity flag, *INTVEL*, on Control Card 7 is blank. Otherwise, define up to *NUMNP* cards as follows.

Cards 1, 2, ... , <i>NUMNP</i>

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-8	Node point number	I8
9-18	Initial velocity in x-direction	E10.0
19-28	Initial velocity in y-direction	E10.0
29-38	Initial velocity in z-direction	E10.0
39-43	Node point increment , <i>k</i> EQ.0: default set to "1"	I5

Nonzero initial velocities may only be specified for dynamic analyses. When included, initial velocity must be defined for each node point. These cards do not need to be in order; however, the highest node point must terminate the data. Whenever data is missing, node numbers are generated according to the sequence $[n_i, (n_i + k), (n_i + 2k), \mathbf{K}, n_j]$, where n_i and n_j are the node numbers defined on two consecutive cards. Linear interpolation is used to obtain the initial velocities of the generated nodes.

4.28 Aerodynamic Drag Load Deck

Skip this section if the number of aerodynamic drag loads, *NUMADL*, on control card 4 is zero.

Card 1

Columns	Quantity	Format
1-5	Load curve giving X-component of free stream velocity versus time EQ. 0: No superimposed velocity in X-direction	I5
6-10	Load curve giving Y-component of free stream velocity versus time EQ. 0: No superimposed velocity in Y-direction	I5
11-15	Load curve giving Z-component of free stream velocity versus time EQ. 0: No superimposed velocity in Z-direction	I5

Cards 2, 3, ... , (<i>NUMADL</i> + 1)
--

Columns	Quantity	Format
1-8	Node point number, n_1	I8
9-16	Node point number, n_2	I8
17-26	Frictional drag multiplier, P	E10.0
27-36	Form drag multiplier, F	E10.0

Aerodynamic drag loads may be applied to beam and truss elements in both steady state and dynamic analyses. They may be used to simulate external forces of magnitude proportional to the velocity of the elements squared, acting in a direction opposite to the velocity. An arbitrary free stream velocity field may be superimposed by specifying the time variation of global components using load curves. Aerodynamic cross sections may be specified independent of mechanical cross sections.

Two drag effects are included. Frictional drag F^P is computed using the segment *surface* area. Form drag F^F is computed using the segment *frontal* area. The scalar parameter λ is computed by NIKE3D from the current segment orientation, and serves to weight the two drag effects. When the segment is oriented normal to the relative velocity vector, λ takes a value of one, and only form drag forces are applied. As the segment inclines with respect to the relative velocity, frictional drag forces are activated. In the limit when the relative velocity is aligned with the segment axis, λ takes a value of zero, and only frictional effects are applied:

$$\mathbf{F}^P = \rho \|\mathbf{V}\|^2 (1 - \lambda) l \mathbf{e}_v$$

$$\mathbf{F}^F = \lambda \|\mathbf{V}\|^2 W \mathbf{e}_v$$

The density of the surrounding fluid, the aerodynamic cross section, and the friction and form drag coefficients are imbedded in the user defined drag multipliers:

$$P = \frac{1}{2} C_D^P W^P$$

$$F = \frac{1}{2} C_D^F W^F$$

where

$$\begin{aligned} \rho &= \text{fluid density,} \\ C_D^P, C_D^F &= \text{frictional and form drag coefficients, respectively,} \\ W^P, W^F &= \text{cross sectional circumference and width, respectively.} \end{aligned}$$

and the quantities computed by NIKE3D are:

$$\begin{aligned} \mathbf{v} &= \text{current relative velocity vector} \\ l &= \text{current segment length} \\ \lambda &= |\mathbf{e}_a \cdot \mathbf{e}_v| \\ \mathbf{e}_a &= \text{unit vector in direction of segment axis} \\ \mathbf{e}_v &= \text{unit vector in direction of relative velocity} \end{aligned}$$

4.29 Foundation Node Boundary Condition

Skip this section if the number of foundation nodes, *NFNBC*, on control card 4 is zero.

Cards 1, 2, ... , (NFNBC)

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-8	Node point number	I8
9-13	Material number (must correspond to a Type #17 material model)	I5

This boundary condition allows a general, 3D, linear spring to be inserted between a node point and the base coordinate system of the model. The 6x6 stiffness matrix relating global displacements and rotations to global forces and moments is entered using material type #17. In this way, several foundation nodes may be conveniently assigned the same spring properties. The stiffness matrix need not be full rank - one or more rows and/or columns may be zero to allow unconstrained motion in selected global directions.

5. RESTART INPUT FORMAT

A restart input deck is generally not needed to restart NIKE3D. It may be used, however, to reset the following parameters:

- analysis type (static or dynamic),
- number of additional time steps,
- time step size,
- output printing interval,
- output plotting interval,
- integration constants,
- equilibrium iteration parameters,
- time histories.

All changes made when restarting will be reflected in the next restart dump. When restarting using the input deck described below, associate the restart input deck file name with the **I=** parameter, and the dump file with the **R=** parameter on the execution line.

5.1 Restart Control Deck

Card 1

Columns	Quantity	Format
1-72	Title card	12A6

Card 2

Columns	Quantity	Format
1-5	Number of additional time steps, n EQ.0: termination time remains unchanged NE. 0: terminate after total number of steps taken in original run plus restarted run equals ($NSTEP + n$), where $NSTEP$ is the number of steps requested in the original run. Termination time for automatic time step control is computed as $T_{new} = T_{old} + n \times dt$	I5
6-15	Time step size, dt	E10.0
16-20	Output printing interval EQ.0: output printing interval remains unchanged	I5
21-25	Output plotting interval EQ.0: plotting interval remains unchanged	I5
26-35	First Newmark integration parameter, Ignored if columns 6-15 are blank; otherwise, defaults to ".50"	E10.0
35-45	Second Newmark integration parameter, Ignored if columns 6-15 are blank; otherwise, defaults to ".25"	E10.0

Card 3

Columns	Quantity	Format
1-5	Analysis type change flag, <i>ITCF</i> EQ.0: no change EQ.1: changed as specified below	I5
6-10	Number of load curves to be redefined, <i>NTHRD</i>	I5
11-15	Number of sliding interfaces to be redefined, <i>NSFRD</i>	I5

Card 4

Columns	Quantity	Format
1-5	Number of time steps between stiffness reformations at beginning of step EQ.0: this parameter remains unchanged	I5
6-10	Number of time steps between equilibrium iterations EQ.0: this parameter remains unchanged	I5
11-15	Maximum number of equilibrium iterations between stiffness matrix reformations EQ.0: this parameter remains unchanged	I5
16-20	Maximum number of stiffness matrix reformations	I5
21-30	Displacement convergence tolerance EQ.0.0: this parameter remains unchanged	E10.0
31-40	Energy convergence tolerance EQ.0.0: this parameter remains unchanged	E10.0
41-50	Residual convergence tolerance EQ.0.0: this parameter remains unchanged	E10.0
51-55	Number of steps between restart file generation EQ.0.0: this parameter remains unchanged	I5

5.2 Analysis Type Change Deck

Skip this card if *ITCF* equals zero, otherwise define one card:

Card 1

Columns	Quantity	Format
1-5	Analysis type EQ.0: static analysis EQ.1: dynamic analysis with lumped mass matrix EQ.2: eigenvalue analysis about deformed geometry	I5
6-10	Number of eigenvalues to extract	I5
11-20	Frequency Shift (cycles per unit time)	I5

In changing from a static to a dynamic analysis, the densities that were defined in the original input deck are used. When restarting for eigenvalue analysis, geometric stiffness terms for brick, beam, and shell elements are automatically activated.

5.3 Load Curve Redefinition Deck

Define *NTHRD* decks, one for each redefined load curve. The total number of points in a load curve, *NPTS*, may not change from the original input.

Card 1

Columns	Quantity	Format
1-5	Load curve number of load curve to be redefined	I5

Card 2,...,NPTS+1

Columns	Quantity	Format
1-10	Time	E10.0
11-20	Load value	E10.0

5.4 Sliding Interface Redefinition Deck

Define *NSFRD* decks, one for each redefined sliding interface. See section 4.12, Sliding Interface Deck, for descriptions of each parameter.

Card 1

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-5	Interface number of interface to be redefined	I5
6-15	Penalty scale factor (multiplies current value)	E10.0
16-25	Static friction coefficient	E10.0
26-35	Dynamic friction coefficient	E10.0
36-45	Friction decay exponent	E10.0
55-55	Small penetration search distance	E10.0

Card 2

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-5	Augmentation flag	I5
6-15	Normal direction convergence tolerance	E10.0
16-25	Tangential direction convergence tolerance	E10.0
26-35	Tangential direction stiffness multiplier	E10.0
36-45	Interface death time	E10.0
46-55	Interface burial time	E10.0

6. EXAMPLES

This section is intended to provide a short demonstration of NIKE3D capabilities, and provide some quantitative results for speed and accuracy comparisons. The first two problems are from the standard NIKE3D test problem set (`n3test1` and `n3test2`). They provide examples which are small enough to run in a reasonable time on workstation class machines. Intermediate results are given for quantitative comparison to workstation results. The remaining three examples demonstrate some of the new features of the code. All plots were produced with the post processor TAURUS.

6.1 Bar Impacting Rigid Wall

This transient dynamic analysis simulates a copper bar of diameter 6.4 mm and length 32.4 mm impacting a rigid wall with an initial axial velocity of 0.227 mm/ μ s. The bilinear elastic-plastic material model is used with isotropic hardening. A 90° segment of this axisymmetric cross section is modeled using 972 solid elements.

The wall is modeled using NIKE3D's stonewall option. The wall could have been modeled explicitly using a layer of solid elements constrained on the bottom surface, with a sliding interface between the top surface and the end of the bar. This approach would have been more costly, since the discretized wall introduces additional (unnecessary) degrees of freedom to the model. Note that this axisymmetric, high rate dynamic analysis is more appropriately a DYNA2D problem.

Table 6-1 gives execution times for the first 20 time steps of the bar impact problem on several computer platforms. Results are shown in figure 6-1 for several stages of deformation. In these plots, the quarter symmetric model was reflected about one symmetry plane using the post processor TAURUS, producing the full cross sections shown. Contours of effective plastic strain are shown in figure 6-2 for an intermediate and final state of deformation. Maximum values exceed 100%, indicating the large degree of material nonlinearity in the model. The final height of the deformed bar at rest was computed to be 20.954 mm. This compares very well to the analytic and experimental values of 19.9 and 21.9 mm, respectively, given by Wilkins and Guinan [1973].

Computer Platform	CPU time (seconds)
Sun Sparc10-51	380
SGI R-4000	306
IBM RS/6000-550	187
Dec Alpha	170
Cray Y/MP	35
Cray C90	19

Table 6-1 Timings for the first 20 steps of the bar impact problem.

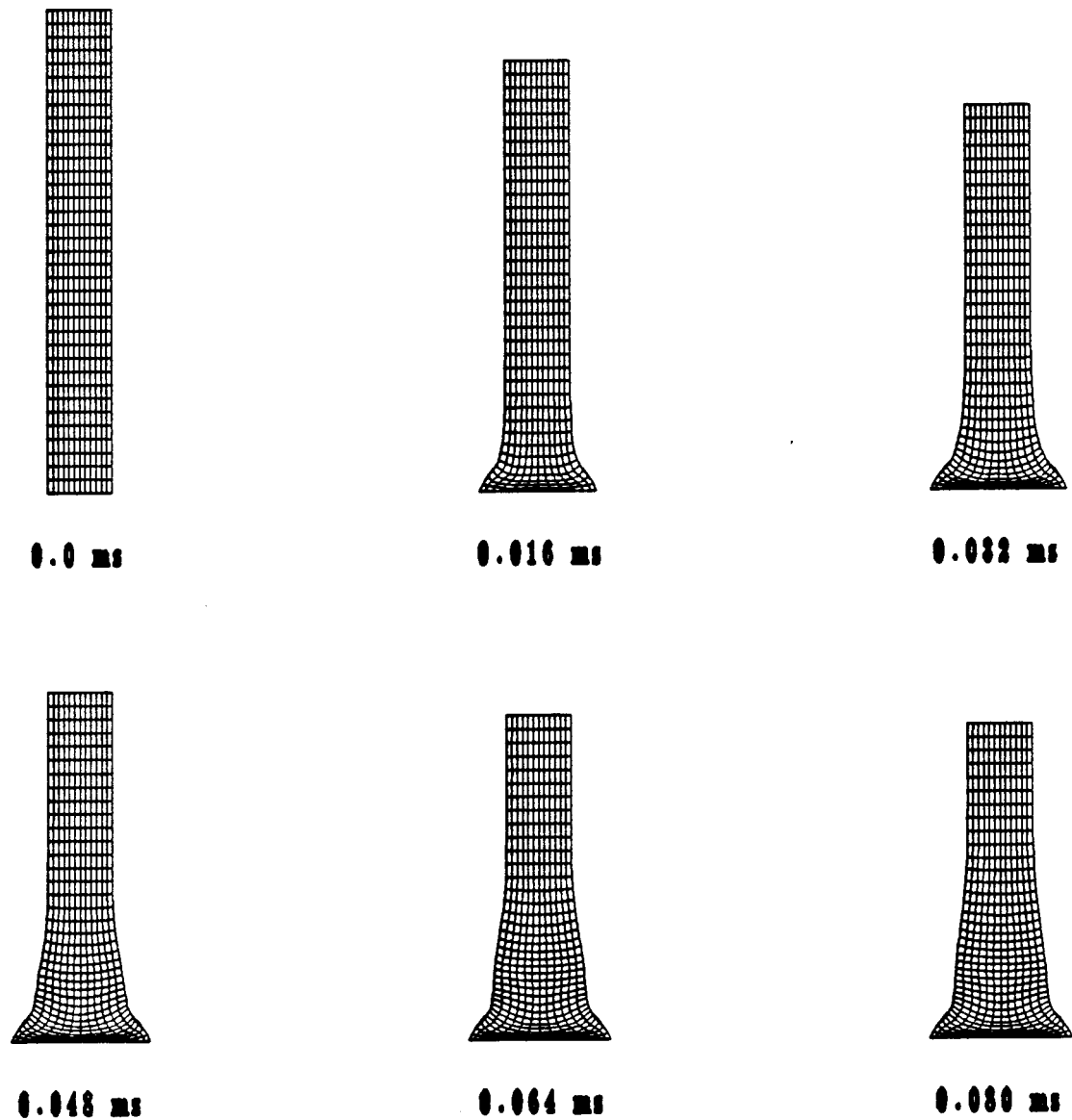


Figure 6-1. Stages of deformation of a copper bar impacting a rigid surface. The final deformed height was computed to be 20.954 mm. This compares very well to the analytic and experimental values of 19.9 and 21.9 mm, respectively.

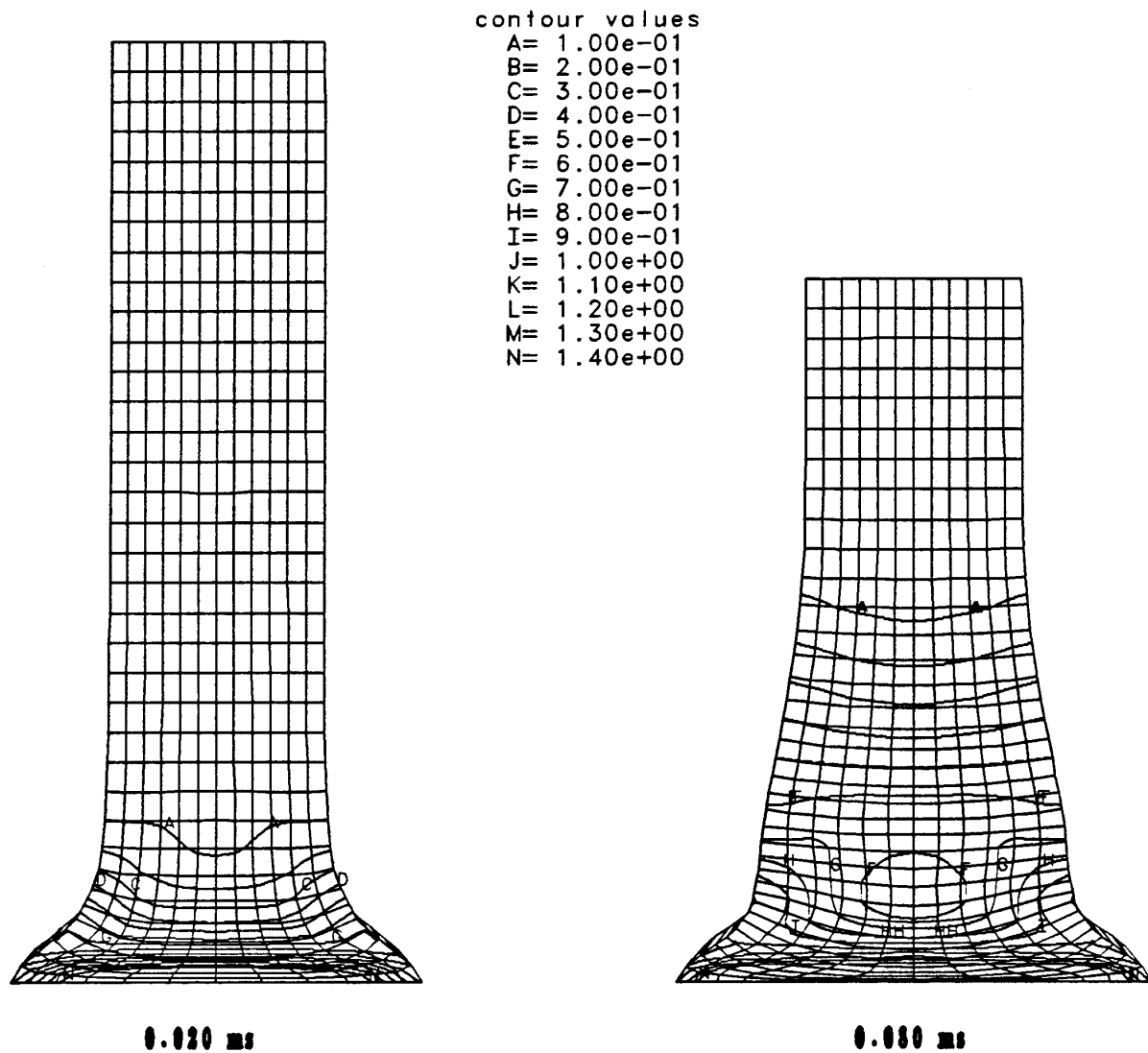


Figure 6-2. Contours of effective plastic strain in the bar impact problem for an intermediate state at time=20 microseconds, and the final state at time=80 microseconds. Maximum values exceed 100% in the final configuration.

6.2 Pipe Whip

This transient dynamic analysis simulates the impact of two steel pipes. The pipes have outside diameter of 3.3125", wall thickness of 0.432", and are 50" long. The target pipe is supported with a fixed boundary condition at each end. The second pipe swings freely about one end with an angular velocity at impact of 50 radians per second. One half of this symmetric problem is modeled. The full problem is displayed during postprocessing using TAURUS' mesh reflection capability.

Shell elements are used in the model. Solid elements could have been used, but since bending behavior of the pipe wall must be represented accurately, several elements would be required in the thickness direction. This would require many more elements in the plane directions to maintain reasonably low aspect ratios (ratio of longest to shortest element side) of even 5:1. Using shell elements results in a much less expensive analysis.

A slide surface (type 3) is included between the pipes. This prevents nodes on one pipe from penetrating element faces of the other pipe, and allows the contact areas to evolve as the pipes deform. Minimizing the total number of slide surface facets reduces the cost of the analysis. Here the slave and master surfaces are defined over half of the pipe circumference, and only at the region of refined mesh. This is judged by intuition to be the minimum potential contact area. If the slave or master surfaces were mistakenly defined over too small an area, physically unrealistic interpenetration would be observed between node/facet pairs not included in the definition.

Computer Platform	CPU time (seconds)
Sun Sparc10-51	338
SGI R-4000	311
IBM RS/6000-550	107
Cray Y/MP	33

Table 6-2 Timings for the first 20 steps of the pipe whip problem.

Table 6-2 gives timing results for the first 20 steps of the pipe whip problem on several computer platforms. Results are shown in figure 6-3 for several stages of deformation. Contact begins at a single point, and evolves to two separate regions. Late in the analysis, a gap opens at the initial contact point. Figure 6-4 shows contours of effective plastic strain on a close up, longitudinal section view of the impacting pipe. At the final stage of deformation, the cross section of the pipe

is severely collapsed, with a local buckle developing at the narrowest point. Observing such severe deformation over very few elements, the experienced analyst may repeat this calculation with a more refined mesh in the impact region to obtain more accurate results at this crucial location.

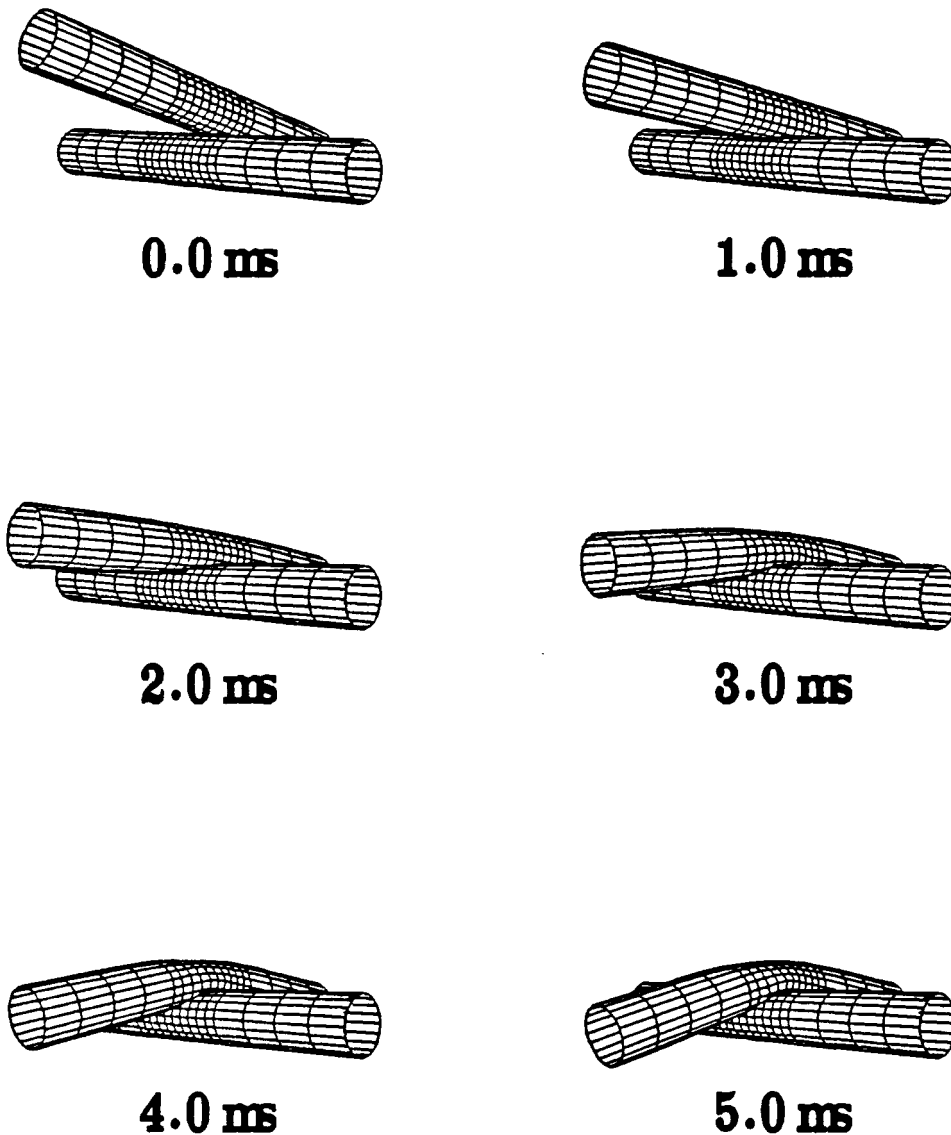


Figure 6-3. Deformed geometry of the pipe whip problem at several stages of the analysis. The lower pipe is clamped at each end, while the upper one swings freely with an initial impact velocity of 50 radians per second. Slide surfaces defined at the refined areas of each pipe allow the contact areas to evolve, and gaps to open as the deformation proceeds.

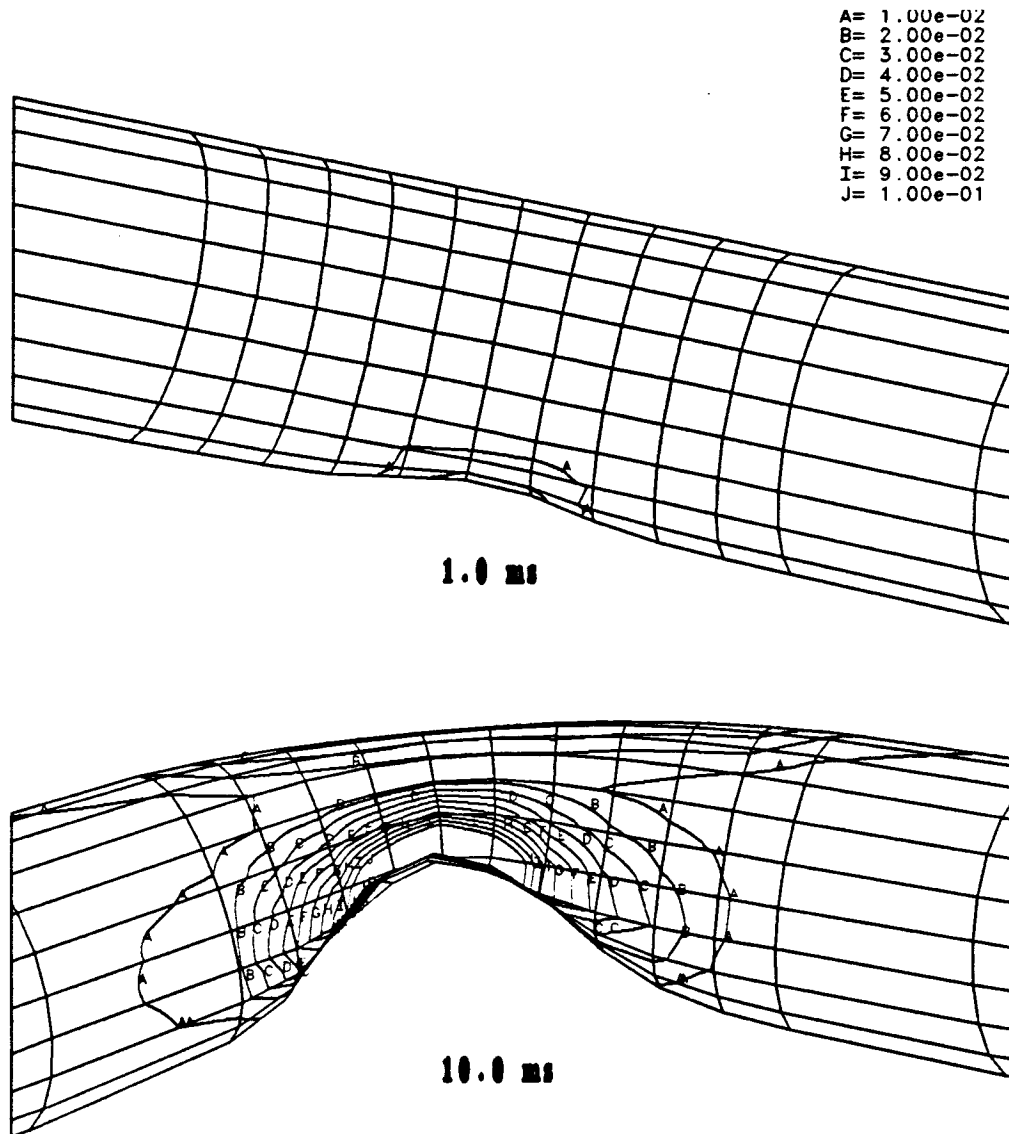


Figure 6-4. Enlarged view of the impacting pipe. Contours of effective plastic strain are plotted on this longitudinal section. After an elapsed time of 1.0 ms, maximum effective plastic strains reach 2% over a relatively small area. At the 10.0 ms mark, values in excess of 10% extend over a large area, centered at the original contact point.

6.3 DYNAMIC RESPONSE OF OVERPASS BRIDGE (D.B. McCallen)

The importance of the nonlinear behavior of civil structures subjected to earthquake loading is becoming more widely recognized. This example summarizes work reported by McCallen and Romstad [1994] on modeling an overpass bridge-soil system using NIKE3D. The detailed bridge model, including discretized soil embankments, is shown in figure 6-5. The individual piles below the abutment pile caps and central column footings are represented explicitly with beam elements. The original grade and embankment soils are modeled with solid elements, using the Ramberg-Osgood material model to represent hysteretic soil behavior. Hughes-Liu shell elements were used to represent the bridge superstructure.

Several analyses were performed using the model. Figure 6-6 shows mode shapes corresponding to the first six natural frequencies of the model as predicted by NIKE3D eigenvalue analysis. By varying soil properties to account for embankment softening during strong ground motion, good correlation with experimental data was obtained over a range of loading conditions.

Measured free field acceleration data from the April 1992 M=7.0 earthquake was applied to the model using NIKE3D's base acceleration boundary condition. Figure 6-7 shows the correlation between measured and computed response at two locations on the bridge. Figure 6-8 shows the deformed model during this simulation, and the nonlinear, hysteretic soil response.

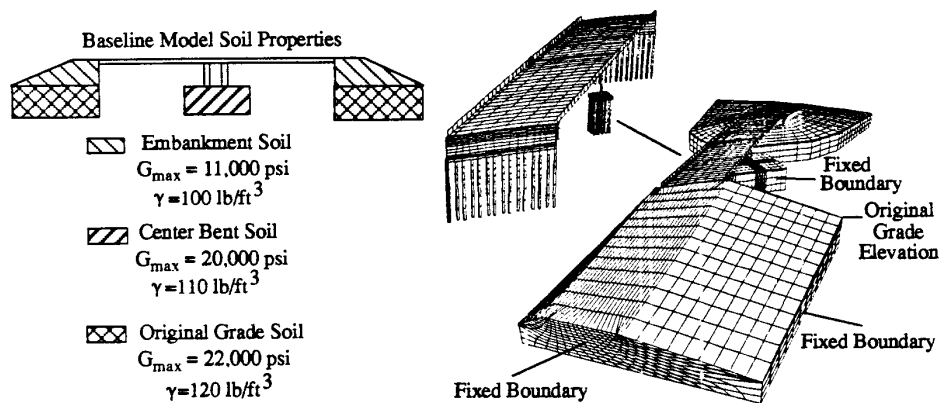


Figure 6-5 Detailed model of bridge/soil system, and properties used in Ramberg-Osgood model of soil.

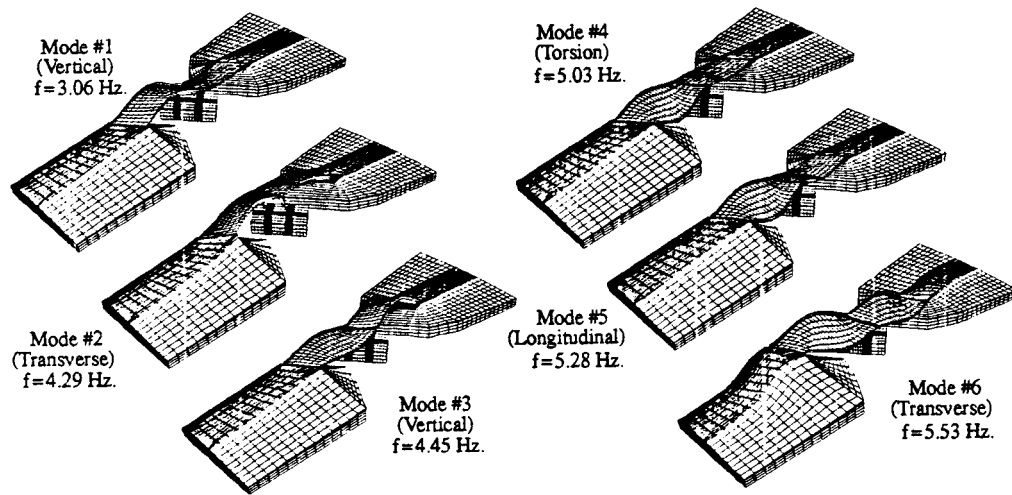


Figure 6-6 First six modes of vibration predicted by NIKE3D.

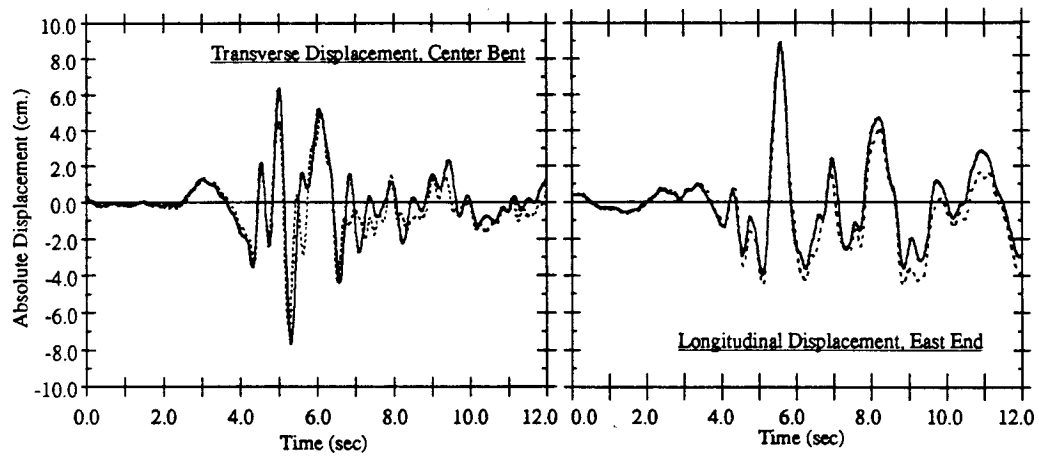


Figure 6-7 Predicted (dashed line) and measured (solid line) response to the April 1992 M=7.0 earthquake.

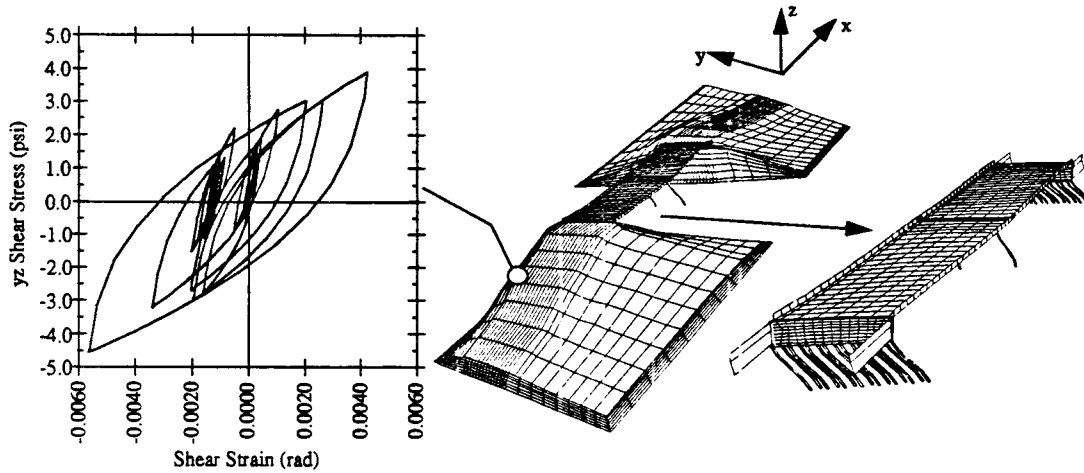


Figure 6-8 Deformed geometry at peak transverse response during earthquake (displacement scale factor = 200) and soil stress-strain history.

6.4 FOOTED COLUMN

The new one-dimensional slideline is highlighted in this example of an elastic column fixed to the ground by four imbedded rods. Each rod is modeled using 20 beam elements, which are constrained to slide internally along the edges of the column's solid elements using the 1D slideline. The column rests on a frictionless surface, the only attachment being at the fully constrained ends of the rods. Under the action of lateral displacements applied at its top, the column tips, extracting the rods in a prying motion. Figure 6-9 shows the system in its fully deformed configuration, viewed from below through the (transparent) frictionless surface.

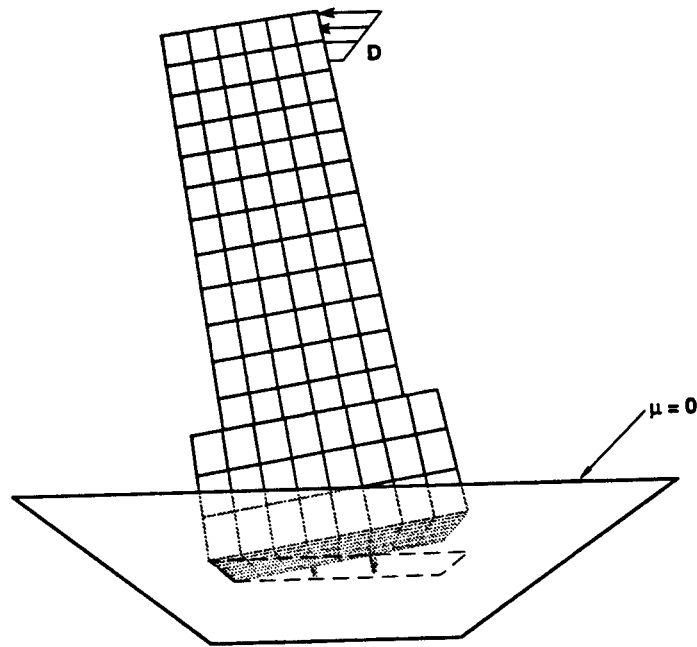


Figure 6-9 Deformed geometry of the footed column. Under the action of applied displacements **D**, the column tips on its frictionless foundation, prying the embedded rods from the column.

The axial slip distance of a particular rod relative to the column is shown in Figure 6-10 for several locations along the length of the rod. Slipping begins at location A, closest to the base of the column, and propagates toward the top of the column, eventually slipping at location D. The total slip distance at location A is nearly ten initial rod element lengths. This results in the emergence of five node points, each punctuated by a slight nonlinearity in the slip/displacement curve. The rod element nearest the top of the column (location D) has a relatively larger diameter, resulting in a greater restraining force, and delayed slipping compared to the other three locations.

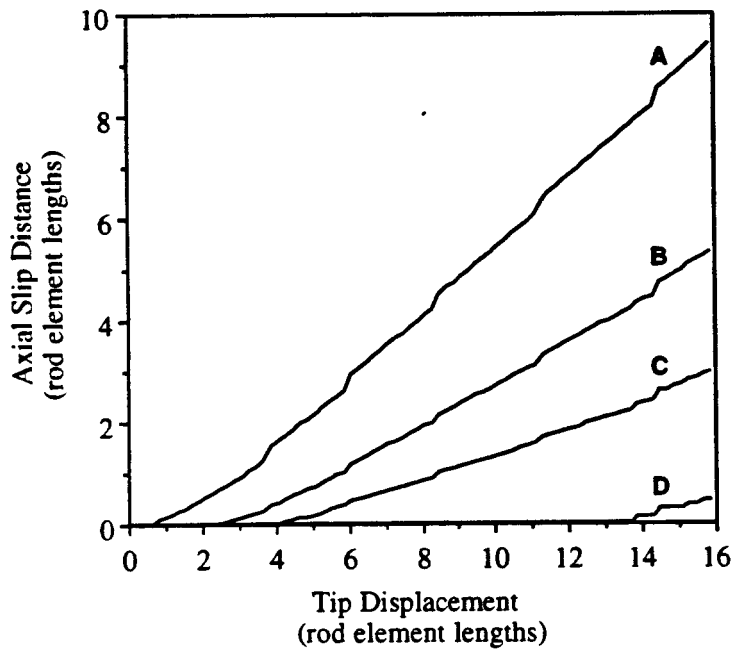


Figure 6-10 Evolution of relative axial slip between rod and column at several locations along the rod, from base (A) to top (D). The rod element at location (D) has a relatively larger diameter, creating increased axial restraining force and delaying the onset of slip.

The stick-slip nature of the axial constraint in the 1D slideline is demonstrated in Figure 6-11, where normalized load in the rod is plotted against the tip displacement of the column for three locations along the rod. The highest load is carried by the rod element closest to the foundation (location A), and this load decreases as the entire rod begins to slip, drawing elements from the column and decreasing the embedded length.

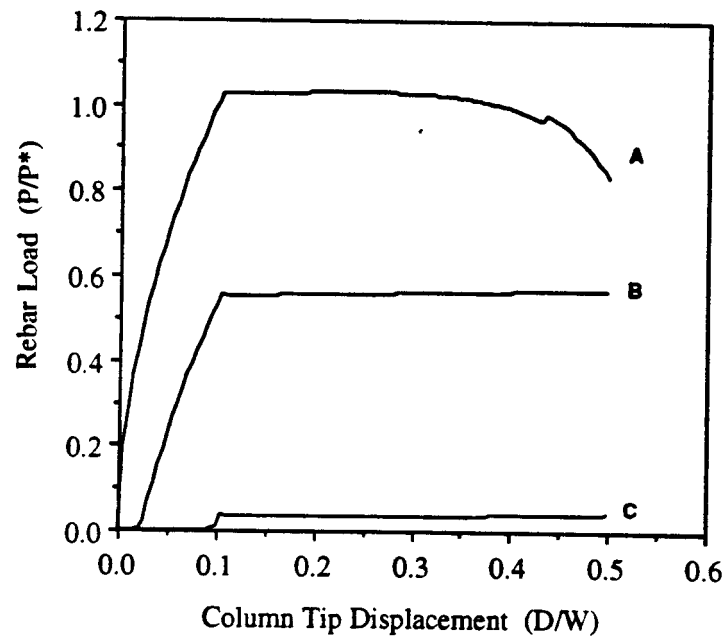


Figure 6-11 Evolution of axial load at several locations along rod, from base (A) to top (C). Load is maximum at the foundation, and eventually decreases as the rod emerges from the column, decreasing the embedded length.

6.5 ELASTIC-PLASTIC TUBE (S.A. Perfect)

This example illustrates a rolling operation, where a metal tube with square cross section is rolled in a three point bending device. Sections of each roller are explicitly modeled using solid elements. Frictionless contact surfaces are defined between the tube and each roller. Figure 6-12 shows a sequence of deformation states, with contours of effective plastic strain. Initially, the leftmost roller is displaced upward, generating plastic strain between it and the middle roller. The right end of the tube is then displaced to the left, pushing the tube through the bending device.

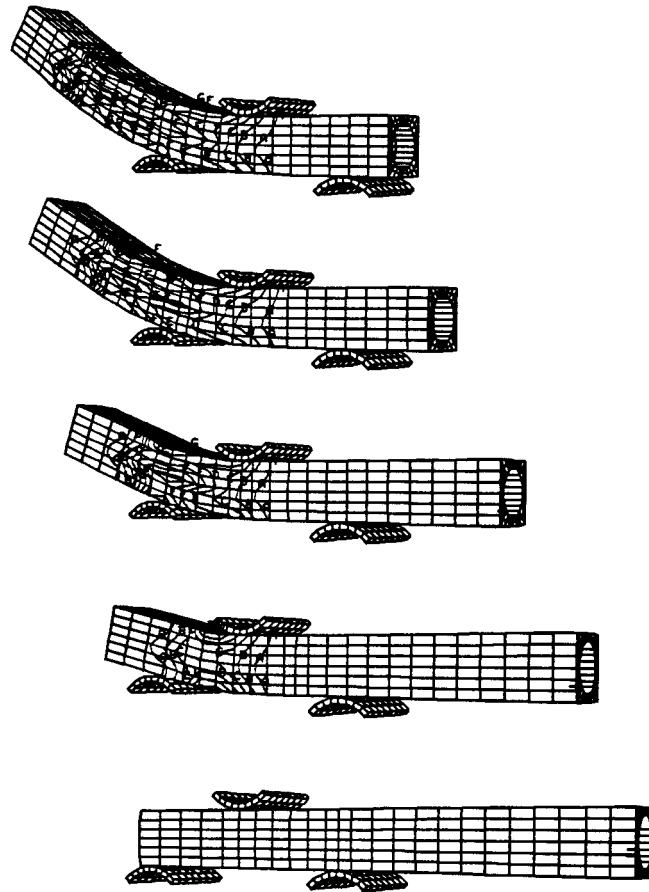


Figure 6-12 Contours of effective plastic strain in a metal tube drawn through a three-point bending device, at various stages of deformation. Contours are drawn from one to seven percent strain, in one percent increments .

6.6 VOID GROWTH IN TENSILE SPECIMEN (P.J. Raboin)

This example highlights the new thermo-plastic melt material in an analysis of a simple tensile specimen (Figure 6-13). Only the elastic-plastic flow strength and Gurson void growth features of this versatile material model were activated. Figure 6-14 shows the growth of voids in the material, plotted against uniaxial strain, for several initial void fractions.

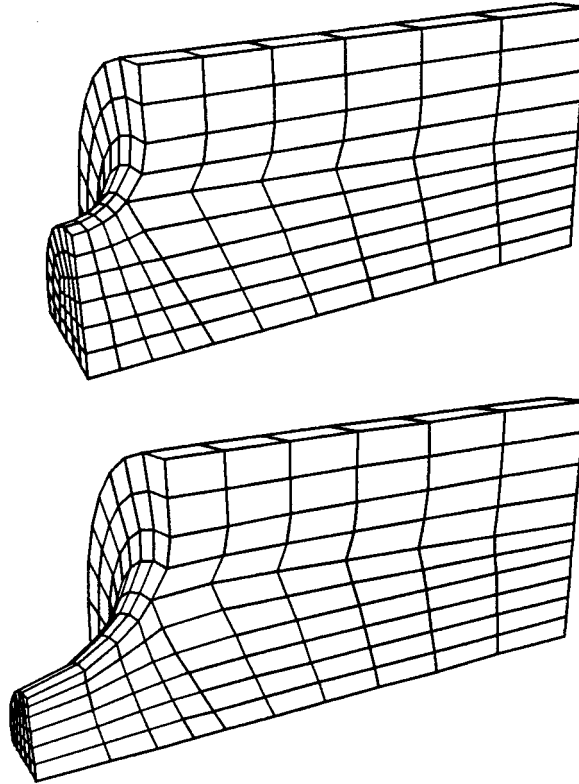


Figure 6-13 Mesh used for the tensile test. One eighth of the dogbone specimen is modeled. Displacement boundary conditions stretch the undeformed mesh (top) to a total uniaxial strain of over 120% at the center of the specimen (bottom).

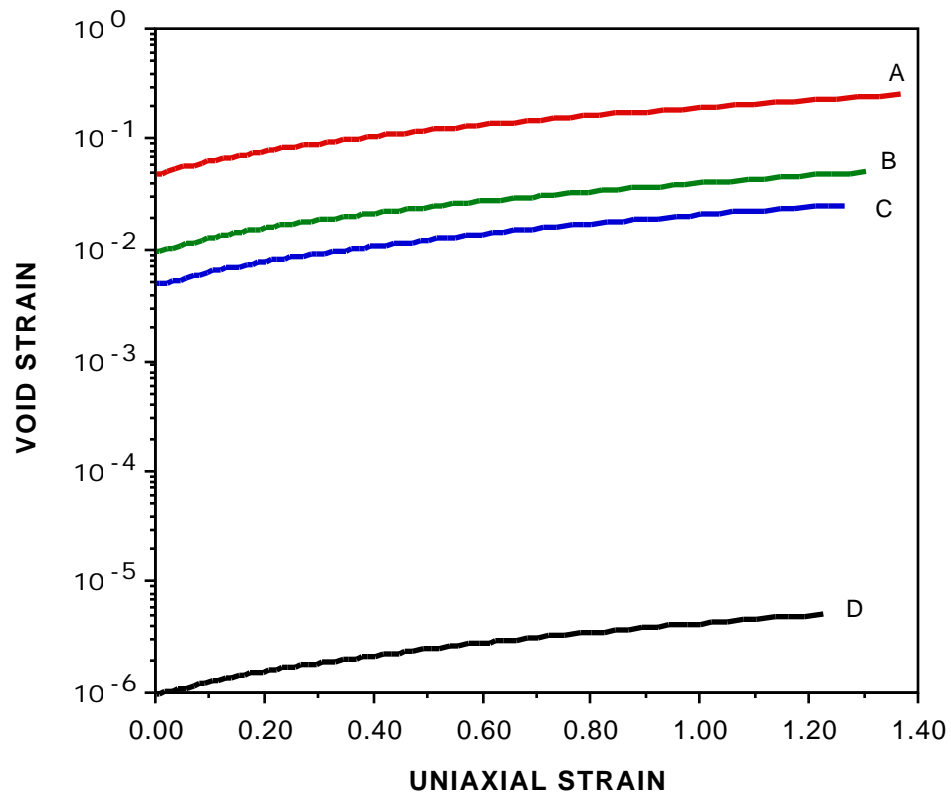


Figure 6-14 Evolution of void strain at the center of the test specimen. Results are given for simulations with initial void strains of 0.0 (A), 0.005 (B), 0.010 (C), and 0.050 (D).

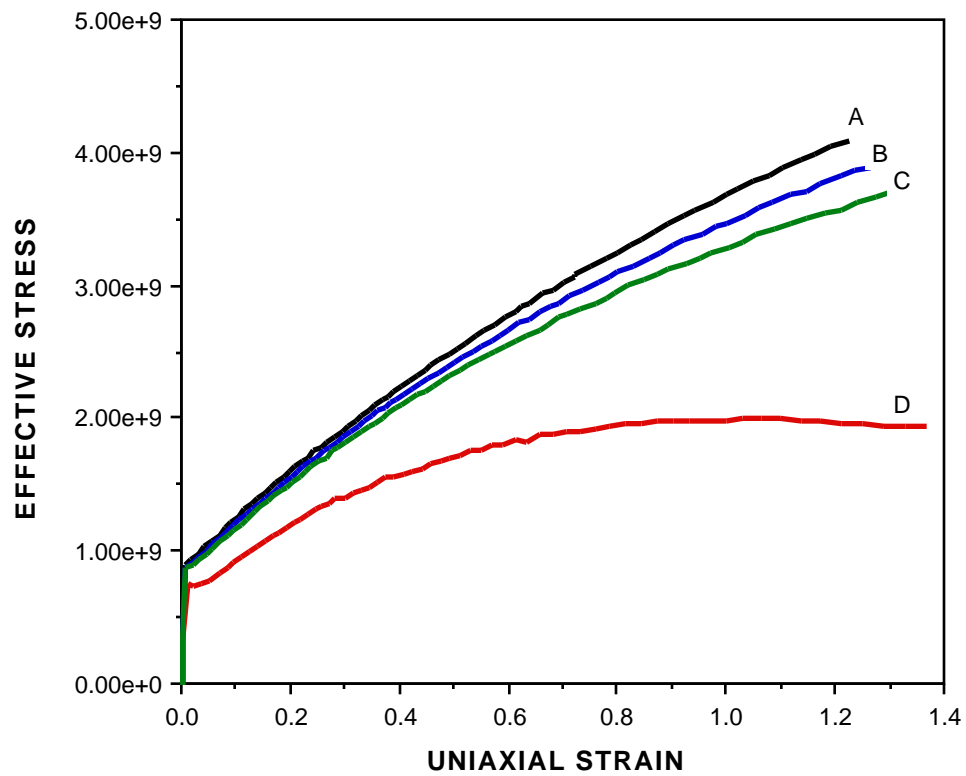


Figure 6-15 Evolution of effective stress at the center of the test specimen. Results are given for simulations with initial void strains of 0.0 (A), 0.005 (B), 0.010 (C), and 0.050 (D).

6.7 HUMAN KNEE JOINT

This example highlights the new rigid and hyperelastic materials in a model of the human knee. The model includes anatomically correct representations of the distal femur, proximal tibia and fibula, collateral and cruciate ligaments, and medial and lateral menisci. All parts were constructed from solid elements. The rigid material model was used to represent the bones, hyperelastic material was used in the menisci, and transversely isotropic hyperelastic material in the ligaments. Figure 6-16 shows the full knee model, and a simplified model of the medial compartment (medial collateral ligament and adjacent bone).

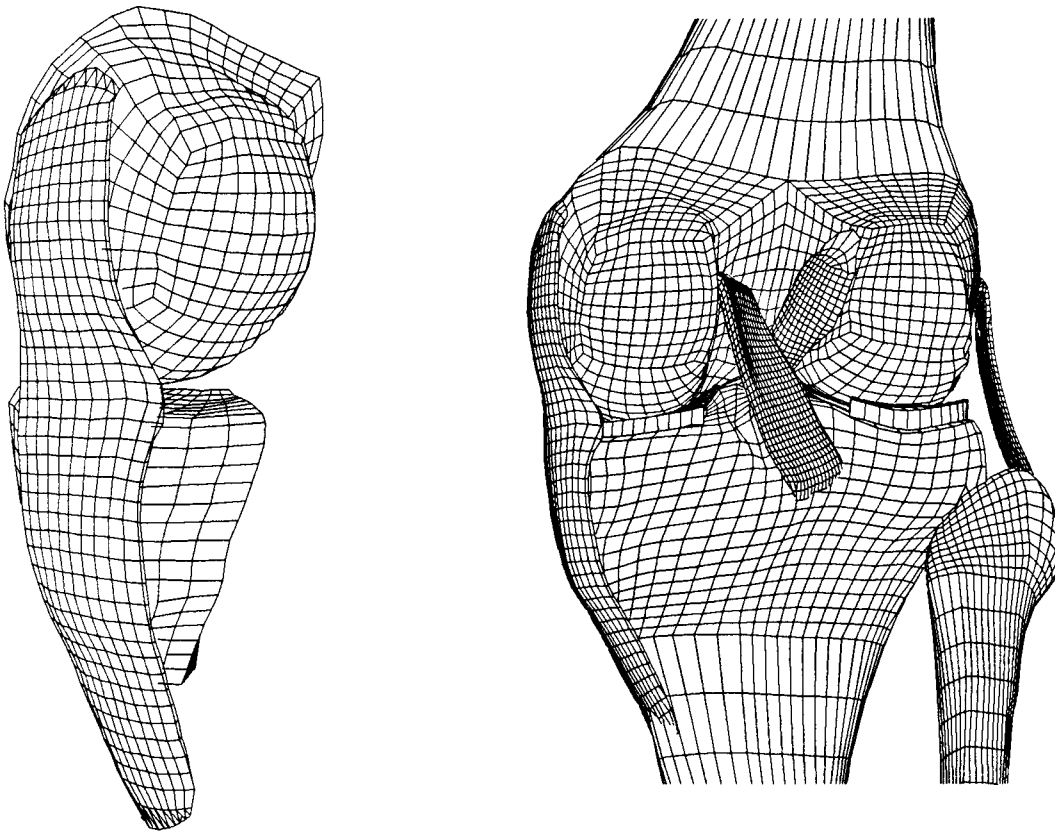


Figure 6-16 The LLNL High Fidelity Knee Model and a simplified medial compartment model.

An important aspect of joint simulation is the application of initial stress to represent the in-vivo state of the soft tissues. Using the medial compartment model with all degrees of freedom of the bones fixed, the initial stretch feature of the transversely isotropic hyperelasticity model was applied to pre-strain the medial collateral ligament to a total load of eight pounds. The femur segment was then rotated 90° to simulate passive joint flexion (figure 6-17).

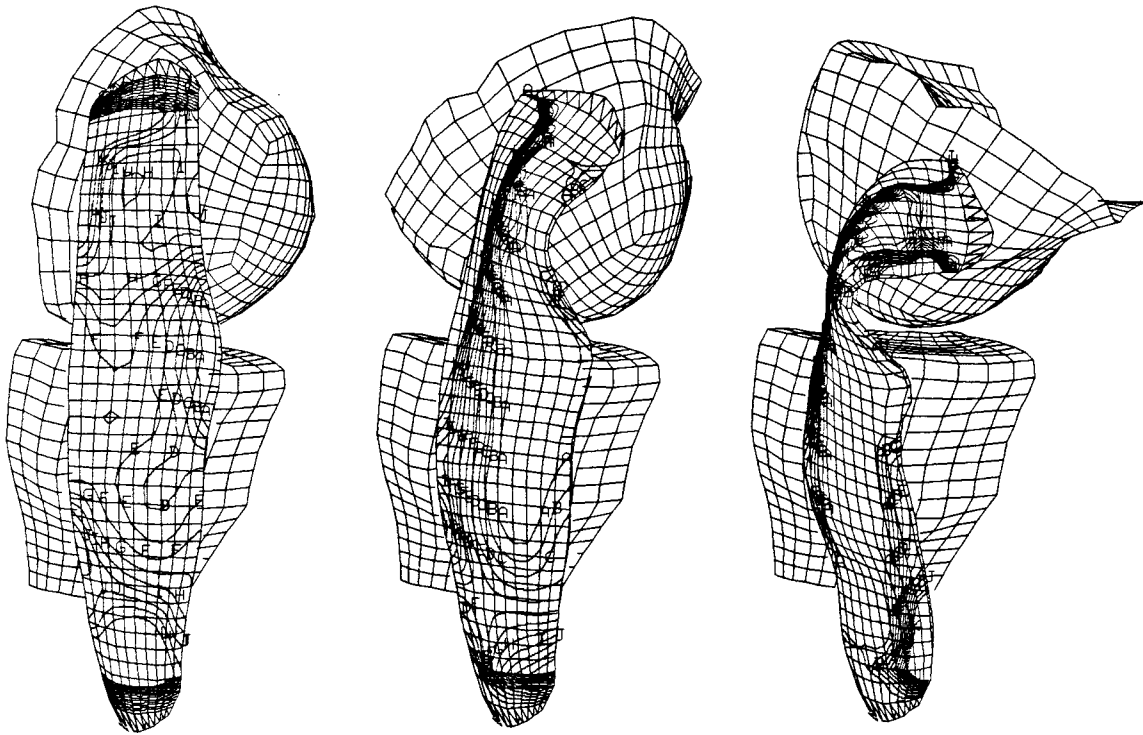


Figure 6-17 Flexion of the knee joint using the simplified medial compartment model causes buckling of the medial collateral ligament. Initial strain in the ligament was adjusted to yield a total load of eight pounds in the pretensioned state (left). This load decreased to six pounds as the joint was flexed to 90 degrees. Contours of maximum principal stress are shown, ranging in values from 60 to 240 psi, in 20 psi increments.

With an appropriate value of initial strain determined, the full knee model was pre-stressed and flexed to a total of 30° hyperextension (figure 6-18). During this simulation, the tibia and fibula were held fixed, and the femur was left free to translate and rotate as necessary to achieve equilibrium in contact with the articular surfaces of the joint, and constrained by the taught ligaments. Only the flexion angle of the joint (femoral rotation) was prescribed. The model predicts maximum stress concentrations in the posterior cruciate ligament in excess of 2500 psi, in agreement with clinical observations of hyperextension injury locations.

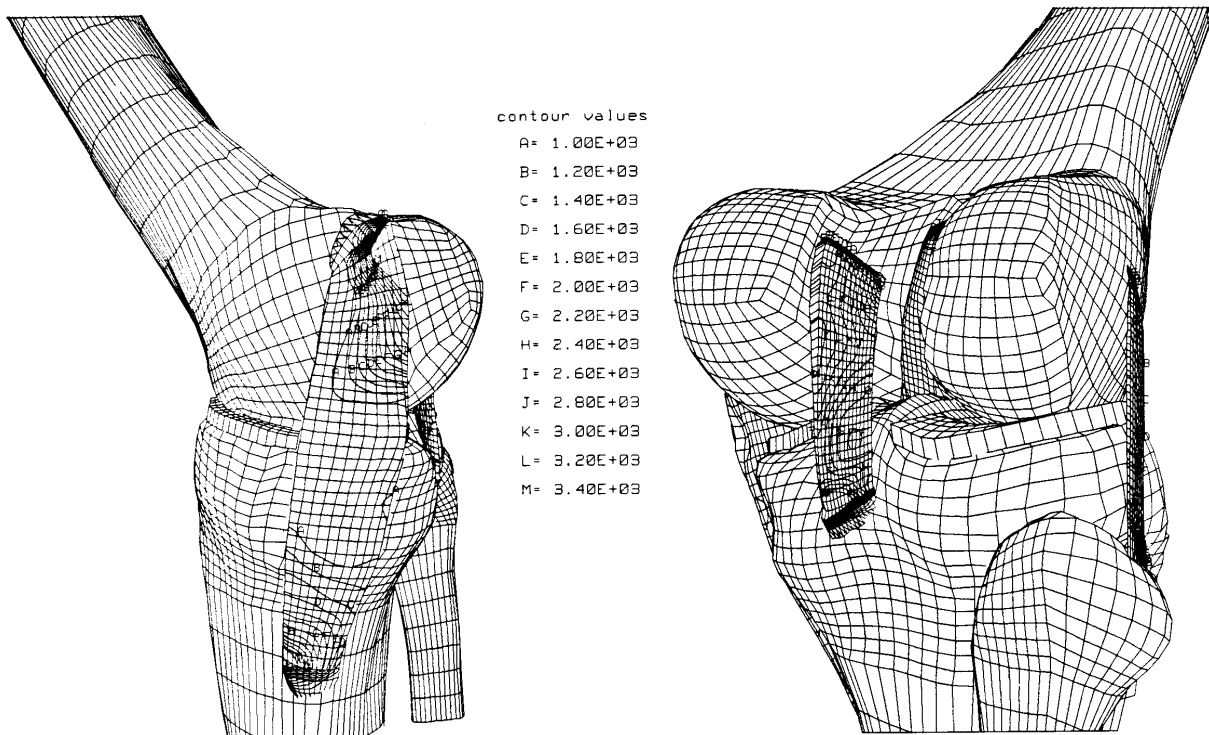


Figure 6-18 The LLNL High Fidelity Knee Model hyperextended to an angle of 30°. After initial ligament pretensioning with the femur free to equilibrate in the joint, counterflexing rotation was prescribed. Contours of maximum principal stress are shown. NIKE3D predicts maximum stresses in excess of 2500 psi in the posterior cruciate ligament as a result of this motion.

7. ACKNOWLEDGMENTS

The author wishes to acknowledge Dr. J.O. Hallquist as the originator of NIKE3D, and author of the kernel of code about which the current version is evolving. The author also wishes to acknowledge Dr. R.M. Ferencz's many contributions, most notably the incompatible modes formulation for solid elements and the element-by-element linear equation solver.

The polar decomposition of the 3 x 3 deformation gradient matrix was derived in closed form by Rodney Stephenson at Carnegie Mellon University and was made available for implementation into NIKE3D by Prof. Thomas J.R. Hughes at Stanford.

Finally, and most importantly, the author would like to acknowledge the NIKE3D development team. This group includes the administration and management personnel who have provided a funding base, the computer support staff, the researchers and consultants who have contributed ideas and formulations, the many code developers who served as sounding boards for algorithm development, and the users both inside and outside LLNL who work to identify and eliminate bugs and demonstrate new features. The success of NIKE3D is a result of the combined efforts of this unique team.

8. REFERENCES

- Bathe**, K.J., and A.P. Cimento, "Some Practical Procedures for the Solution of Nonlinear Finite Element Equations," *Comp. Meths. Appl. Mechs. Eng.*, **22**, pp. 59-85, **1980**.
- Bathe**, K.J. and E.L. Wilson, Numerical Methods in Finite Element Analysis, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, **1976**.
- Belytschko**, T.B. and C.S. Tsay, "Explicit Algorithms for Nonlinear Dynamics of Shells," *Comp. Meths. Appl. Mechs. Eng.*, **43**, pp. 251-276, **1984**
- Cook**, R.D. Concepts and Applications of Finite Element Analysis, John Wiley and Sons, Inc., **1974**.
- Crisfield**, M.A., "A Fast Incremental/Iterative Solution Procedure that Handles Snap Through", *Computers and Structures*, **13**, pp. 55-62, **1981**
- Dovey**, D., and T.E. Spelce, "GRIZ: Finite Element Analysis Results Visualization for Unstructured Grids - User Manual," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-MA-115696, **1993**.
- Engelmann**, B.E. and J.O. Hallquist, "NIKE2D: A Nonlinear, Implicit, Two-Dimensional Finite Element Code For Solid.Mechanics - User's Manual", University of California, Lawrence Livermore National Laboratory, Rept. forthcoming, **1991**
- Engelmann**, B.E., R.G. Whirley, and G.L. Goudreau, " A Simple Shell Element Formulation for Large-Scale Elastoplastic Analysis," in Analytical and Computational Models of Shells, CED-Vol 3, The American Society of Mechanical Engineers, New York, **1989**
- Ferencz**, R.M., " Element-By-Element Preconditioning Techniques for Large-Scale, Vectorized Finite Element Analysis in Nonlinear Solid and Structural Mechanics," Ph.D. Thesis, Stanford University, **1989**
- Gerhard**, M.A., "SLIC: An Interactive, Graphic Mesh Generator for Finite- Element and Finite Difference Application Programs," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52823, **1979**.
- Gibbs**, N.E., W.G. Poole Jr., and P.K. Stockmeyer, "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix," *SIAM J. Num. Anal.*, Vol. 13, **1976**.
- Govindjee**, S., G.J. Kay, and J.C. Simo, "Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete," submitted to: *Int. J. Num. Meth. Eng.*, **1992**
- Hallquist**, J. O., "Theoretical Manual for DYNA3D," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19401, **1983**.
- Hallquist**, J. O., "NIKE3D: An Implicit, Finite-Deformation, Finite Element Code for Analyzing the Static and Dynamic Response of Three-Dimensional Solids," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18822, Rev. 1, **1984**.
-

-
- Hallquist, J. O.**, "DYNA3D Course Notes," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19899, Rev. 2, **1987**.
- Hughes, T.J.R.**, Generalization of Selective Integration Procedures to Anisotropic and Nonlinear Media, *Int. J. Num. Meth. Eng.*, Vol. 15, No. 9, pp. 1413-1418, **1980**.
- Hughes, T.J.R.**, and E. Carnoy, "Nonlinear Finite Element Shell Formulation Accounting for Large Membrane Strains," *Nonlinear Finite Element Analysis of Plates and Shells*, AMD-Vol. 48, American Society of Mechanical Engineers, pp. 193-208, **1981a**.
- Hughes, T.J.R.**, and W.K. Liu, "Nonlinear Finite Element Analysis of Shells: Part I. Three-Dimensional Shells," *Comp. Meths. Appl. Mechs. Eng.*, **26**, pp. 331-362, **1981b**.
- Hughes, T.J.R.**, and W.K. Liu, "Nonlinear Finite Element Analysis of Shells: Part II. Two-Dimensional Shells," *Comp. Meths. Appl. Mechs. Eng.*, **27**, pp. 167-181, **1981c**.
- Hughes, T.J.R.**, W. K. Liu, and I. Levit, "Nonlinear Dynamics Finite Element Analysis of Shells," *Nonlinear Finite Element Analysis in Structural Mechanics*, Editors: W. Wunderlich, E. Stein, and K. J. Bathe, Springer-Verlag, Berlin, pp. 151-168, **1981d**.
- Johnson, G.C.** and D.J Bammann, "A Discussion of Stress Rates in Finite Deformation Problems," *Int. J. Solids Structures*, **20**, 8, pp. 725-737, **1984**
- Key, S.W.**, "HONDO - A Finite Element Computer Program for the Large Deformation Dynamic Response of Axisymmetric Solids," Sandia National Laboratories, Albuquerque, NM, Rept. 74-0039, **1974**.
- Krieg, R.D.**, "Numerical Integration of Some New Unified Plasticity - Creep Formulations," SMiRT-4, M6/4, **1977**.
- Krieg, R.D.**, "A Simple Constitutive Description for Soils and Crushable Foams," SC-DR-72-0883, Sandia Laboratories, Albuquerque, New Mexico, **1972**
- Krieg, R.D.** and S.W Key, Implementation of a Time Dependent Plasticity Theory into Structural Computer Programs, Vol. 20 of *Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects*, American Society of Mechanical Engineers, New York, pp. 125- 137, **1976**
- Laursen, T.A.** and B.N. Maker, "An Augmented Lagrangian Quasi-Newton Solver for Constrained Nonlinear Finite Element Applications" *Int. J. Num. Meth. Eng.*, (to appear), **1995**
- Maker, B.N.** and T.A. Laursen, "A Finite Element Formulation for Rod-Continuum Interactions: The One-Dimensional Slideline," *Int. J. Num. Meth. Eng.*, **37**, 1, pp 1-18, **1994**.
- Maker, B.N.**, "Rigid Bodies for Metal Forming Analysis with NIKE3D," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-JC-119862, **1995**
- Matthies, H.** and G. Strang, "The Solution of Nonlinear Finite Element Equations," *Int. J. Num. Meth. Eng.*, **14**, 11, pp 1613-1626, **1979**.
- McCallen, D.B.** and K.M. Romstad, "Dynamic Analyses of Skewed Short-Span, Box-Girder Overpass," *Earthquake Spectra.*, **10**, 4, p. 729, **1994**.
-

- Murray**, R.C., "Gemini - A Computer Program for Two and Three Dimensional Linear Static, And Seismic Structural Analysis - CDC 7600 Version," University of California, Lawrence Livermore National Laboratory, UCID-20338, p. 40, **1984**
- Nagtegaal**, J.C., D.M. Parks, and J.R. Rice, "On Numerically Accurate Finite Element Solutions in the Fully Plastic Range," *Comp. Meths. Appl. Mechs. Eng.*, 4, 153, **1974**.
- Oliver**, J., "A Consistent Characteristic Length of Smeared Cracking Models," *Int. J. Num. Meth. Eng.*, **28**, pp 461-474, **1989**.
- Puso**, M., Weiss, J.A., "Finite element implementation of anisotropic quasilinear viscoelasticity using a discrete spectrum approximation," *ASME Journal of Biomechanical Engineering*., to appear, **1996**.
- Raboin**, P.J., "A Solidification Model for NIKE2D and NIKE3D" University of California, Lawrence Livermore National Laboratory, UCRL-ID-116849, **1994**
- Ramm**, E., "Strategies for Tracing Nonlinear Response Near Limit Points," Europe-U.S. Workshop: Nonlinear Finite Element Analysis in Structural Mechanics, Eds. Bathe, Stein, and Wunderlich, Springer-Verlag, Berlin, pp. 63-89, **1980**
- Sackett**, S.J. private communication, University of California, Lawrence Livermore National Laboratory, **1977**.
- Shapiro**, A.B., "TOPAZ3D - A Three Dimensional Finite Element Heat Transfer Code," University of California, Lawrence Livermore National Laboratory, Rept. UCID-20484, **1985**
- Simo**, J.C., and R.L. Taylor, "Quasi-Incompressible Finite Elasticity in Principal Stretches: Continuum Basis and Numerical Algorithms," *Comp. Meths. Appl. Mechs. Eng.*, **85**, pp. 273-310, **1991**
- Slater**, J.H., "Mixed Model Finite Elements for Inelastic Analysis," Ph.D. Thesis, University of California, Berkeley, **1982**.
- Spelce**, T., J.O. Hallquist, and B.E. Brown, "TAURUS: An Interactive Post-Processor for the Analysis Codes NIKE3D, DYNA3D, TOPAZ3D, and GEMINI," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-MA-105401, **1990**.
- Stillman**, D.W. and J.O. Hallquist, "INGRID - A Three Dimensional Mesh Generator for Modeling Nonlinear Systems," University of California, Lawrence Livermore National Laboratory, Rept. UCID-20506, **1985**.
- Storassli**, O.O., D.T. Nguyen, and T.K. Agarwal, "A Parallel-Vector Algorithm for Rapid Structural Analysis on High-Performance Computers," NASA Technical Memorandum 102614, Langley Research Center, Hampton, VA, 23665-5225, **1990**.
- Taylor**, R.L., P.J. Beresford, and E.L. Wilson, "A Non-Conforming Element for Stress Analysis," *Int. J. Num. Meth. Eng.*, **10**, 1211-1219, **1976**.
- Taylor**, R.L., private communication, U. C. Berkeley, Department of Civil Engineering, **1979**.
-

- Taylor**, R.L., E.L. Wilson, and S.J. Sackett, "Direct Solution of Equations By Frontal and Variable Band Active Column Methods," Europe-U.S. Workshop: Nonlinear Finite Element Analysis in Structural Mechanics, Eds. Bathe, Stein, and Wunderlich, Springer-Verlag, Berlin, pp. 63-89, **1980**
- Walker**, H.F., "Numerical Solution of Nonlinear Equations," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18285, **1979**.
- Weiss**, J.A. "A Constitutive Model and Finite Element Representation for Transversely Isotropic Soft Tissues," Ph.D. Thesis, Department of Bioengineering, The University of Utah, **1994**.
- Whirley**, R.G. and J.O. Hallquist, "DYNA3D: A Nonlinear, Explicit, Three-Dimensional Finite Element Code For Solid And Structural Mechanics - User's Manual," University of California, Lawrence Livermore National Laboratory, Rept. forthcoming, **1991**.
- Whirley**, R.G. and G.A. Henshall, "Creep Deformation Structural Analysis Using an Efficient Numerical Algorithm," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-JC-104747, **1990**.
- Wilkins**, M.L. and M.W. Guinan, "Impact of Cylinders on a Rigid Boundary," *J. of App. Physics*, **44**, 1973